Spring 2015

Models predicting the performance of IC component or PCB channel during electromagnetic interference

Chunchun Sui

Department: Electrical and Computer Engineering

MODELS PREDICTING THE PERFORMANCE OF IC COMPONENT OR PCB CHANNEL DURING ELECTROMAGNETIC INTERFERENCE

by

CHUNCHUN SUI

A DISSERTATION

Presented to the Faculty of the Graduate School of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

DOCTOR OF PHILOSOPHY

in

COMPUTER ENGINEERING

2015

Approved by

Dr. Daryl Beetner, Advisor
Dr. David J. Pommerenke
Dr. Jun Fan
Dr. Yiyu Shi
Dr. Dan Lin

PUBLICATION DISSERTATION OPTION

This dissertation has been prepared in the style utilized by the Journal of IEEE (Institute of Electrical and Electronics Engineers) Transactions on Electromagnetic Capability, listed as follows:

Paper 1, C. Sui, S. Bai, T. Zhu, C. Cheng, D. Beetner, "New Methods to Characterize Deterministic Jitter and Crosstalk Induced Jitter from Measurements", accepted by IEEE Trans. on EMC., 2015.

Paper 2, C. Sui, L. Ren, X. Gao, X. Li, J. Drewniak, D. Beetner, "Predicting Statistical Characteristics of Jitter Due to Simultaneous Switching Noise", submitted to IEEE Trans. on EMC., 2015.

Paper 3, C. Sui, X. Gao, D. J. Pommerenke, D. Beetner, "Modeling Delay Variations of DLL Due to Power and Ground Voltage Fluctuations", to be submitted to IEEE Trans. on EMC., 2015.

ABSTRACT

This dissertation is composed of three papers, which cover the prediction of the characteristics of jitter due to crosstalk and due to simultaneous switching noise, and covers susceptibility of delay locked loop (DLL) to electromagnetic interference.

In the first paper, an improved tail-fit de-convolution method is proposed for characterizing the impact of deterministic jitter in the presence of random jitter. A Wiener filter de-convolution method is also presented for extracting the characteristics of crosstalk induced jitter from measurements of total jitter made when the crosstalk sources were and were not present. The proposed techniques are shown to work well both in simulations and in measurements of a high-speed link.

In the second paper, methods are developed to predict the statistical distribution of timing jitter due to dynamic currents drawn by an integrated circuit (IC) and the resulting power supply noise on the PCB. Distribution of dynamic currents is found through vectorless methods. Results demonstrate the approach can rapidly determine the average and standard deviation of the power supply noise voltage and the peak jitter within 5~15% error, which is more than sufficient for predicting the performance impact on integrated circuits.

In the third paper, a model is developed to predict the susceptibility of a DLL to electromagnetic noise on the power supply. With the proposed analytical noise transfer function, peak to peak jitter and cycle to cycle jitter at the DLL output can be estimated, which can be use to predict when soft failures will occur and to better understand how to fix these failures. Simulation and measurement results demonstrate the accuracy of the DLL delay model.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to Dr. Daryl Beetner, my advisor, for his guidance and instruction on my research work, financial support to my study and direction for this dissertation during my pursuit of the PhD's degree. I learned not only academic knowledge from Dr. Beetner, but also the rigorous attitude, good manners, and courage to face difficulties.

I would like to thank Dr. Jun Fan, Dr. David J. Pommerenke, Dr. Victor Khilkevich and Dr. James Drewniak, for their teaching in my courses, discussions related to my research and helpful suggestions on my dissertation.

I would like to thank Dr. Minsu Choi, Dr. Yiyu Shi, Dr. Dan Lin, my previous advisor and committee members, for the discussion and helpful suggestions on my research.

I would also like to express my thanks to all the other faculty members and students in and out of the EMC lab for their team work and help in my coursework and research.

Finally, I would like to thank my family for their endless love and endless support in my life.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Figure Page

PAPER I

PAPER II

PAPER III

LIST OF TABLES

1. INTRODUCTION

For high speed circuits, a small amount of crosstalk can eat up the jitter budget and create timing issues. Precise crosstalk jitter characterization of signals at critical internal nodes provides valuable information for hardware fault diagnosis and next generation design. Understanding how crosstalk jitter contributions to overall jitter is challenging, because many types of jitter are combined together. One goal of this dissertation is to evaluate the crosstalk jitter contribution in a high speed I/O link. In this dissertation, crosstalk jitter and random jitter are modeled in simulation. A new tail-fit method is proposed to estimate the probability distribution for random jitter. This new tail fit method is applicable for the decomposition of the crosstalk jitter and random jitter. After prediction characteristics of the random jitter, de-convolution algorithm need to be implemented to separate the random jitter contribution and crosstalk jitter contribution in the total jitter histogram. The result of de-convolution is significantly impacted by noise. The noise is from the total jitter histogram measurement and also from the random jitter prediction. A de-convolution method is needs to be developed that works well in the presence of this noise.

Advanced high-speed I/O interconnects raise considerable signal integrity issues, e.g. switching power supply. An increase of switching power supply noise induces an increase in jitter which may cause severe degradation of the timing margin. Prediction of switching noise induced jitter can help to improve the system's timing margin and achieve lower error rate designs. One goal of this dissertation is to predict the statistical characteristics of jitter from an estimate of the statistical characteristics of switching noise. Current and future works on this topic include predicting the statistical distribution

of power ground noise due to simultaneous switching in FPGAs, predicting the statistical characteristics of jitter based on the predicted power ground noise, and validating the prediction method in simulation and measurement.

Generation and distribution of clock signals inside the IC is critical to the function of an IC. If the clock jitter is sufficiently large, it will cause timing and functional issue in the IC. Delay-locked loops (DLLs) are widely used in multiphase clock generators, clock de-skewing circuits and clock recovery circuits. In the previous researchers' work, there is no analytical equation described the DLL delay variation due to power/ground voltage fluctuations, like those occur during an Electrical Fast Transient (EFT), Electromagnetic Pulse (EMP), or High Power Microwave (HPM) event. One goal of this dissertation is to develop an analytical delay/jitter model of the DLL, which can describe the output phase variation due to power/ground voltage fluctuations.

This dissertation consists of three papers which focus on models predicting the performance of IC component or PCB channel during electromagnetic interference. Paper 1 proposes a method was proposed to estimate the pdf of random jitter from measurements of total jitter. This result can then be used to estimate the contribution of deterministic jitter (which includes crosstalk induced jitter). A method was also proposed for estimating crosstalk induced jitter from measurements of total jitter. Paper 2 proposes a methodology was proposed using vectorless methods to estimate the statistical characteristics of peak power supply noise and peak jitter due to power supply noise. Paper 3 proposes an analytical delay model of VCDL was proposed to predict propagation delay variations when the power supply is disturbed by an electromagnetic event.

The primary contributions of this dissertation include:

Improve the characterization of the crosstalk jitter contribution in the channel. (paper 1).

Vectorless prediction of jitter distribution before circuits design. (paper 2).

Rapid evaluation of circuits performance based on the characteristics of jitter. (paper 2).

Better understanding the delay variation of the DLL circuits. (paper 3).

Simple predictive model for DLL/PLL jitter. (paper 3).

PAPER

I. New Methods to Characterize Deterministic Jitter and Crosstalk Induced Jitter from Measurements

Chunchun Sui, *Student Member, IEEE,* Siqi Bai, Ting Zhu, Christopher Cheng, Daryl G. Beetner, *Senior Member, IEEE*

Abstract—A small amount of jitter can quickly eat up timing budgets and create timing issues. Precise characterization of deterministic and crosstalk induced jitter can help isolate and solve issues within high-speed links. Characterizing deterministic and crosstalk induced jitter is challenging, however, because many types of jitter work together to create the overall jitter profile. Methods are presented in this paper to characterize the deterministic and crosstalk induced jitter from measurements of total jitter. An improved tail-fit de-convolution method is proposed for characterizing the impact of deterministic jitter in the presence of random jitter. The contribution of random jitter to total jitter is found first, and then that contribution is accounted for to find deterministic jitter. A Wiener filter de-convolution method is also presented for extracting the characteristics of crosstalk induced jitter from measurements of total jitter made when the crosstalk sources were and were not present. The Wiener filter allows for accurate deconvolution of the measured histograms for total jitter even in the presence of measurement noise. The proposed techniques are shown to work well both in simulations and in measurements of a high-speed link.

Index Terms— Crosstalk induced jitter, Jitter decomposition, De-convolution, Wiener filter

I. INTRODUCTION

Understanding the causes of jitter is critical to mitigating signal integrity issues in many high-speed digital circuits. Jitter can degrade the timing margin and cause functional issues in the circuit. Jitter is caused by a variety of mechanisms, such as channel loss and reflection, random noise, crosstalk, and other noise. Understanding the contribution of each of these sources of jitter is challenging, since multiple mechanisms work together to generate the total jitter. Precise characterization of the jitter can be a critical tool, however, for improving the circuit design. This paper focuses on the modeling and characterization of deterministic and crosstalk-induced jitter.

The sources of jitter are illustrated in Fig. 1 [1]. Total jitter includes deterministic jitter and random jitter. Random jitter is caused by random noise in the system and generally follows a Gaussian probability density function (pdf). The amplitude of random jitter is unbounded. Deterministic jitter is caused by imperfections in devices or by crosstalk or other circuit issues. It often has a non-Gaussian pdf. There are three types of deterministic jitter: data-dependent jitter, periodic jitter and bounded uncorrelated jitter.

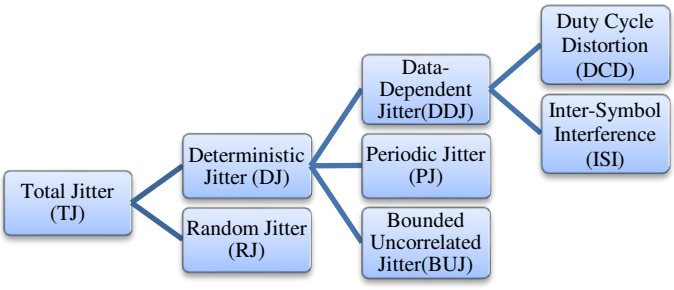

Fig. 1. Common types of jitter

Data-dependent jitter is caused by duty cycle distortion and inter-symbol interference. Duty-cycle distortion results when driver characteristics, such as switching

threshold, bias, or supply voltage, cause the duration of a logical ‘1’ to be different than the duration of a logical ‘0’. Duty-cycle distortion jitter can be modeled by the dual-Dirac delta function [2]. Inter-symbol interference is usually caused by bandwidth limitations of transmission lines, and depends on both data pattern and on the channel or medium system response function.

Periodic jitter, sometimes also called sinusoidal jitter, repeats at a predictable period and is typically caused by deterministic interference, such as from switching power supply noise or a strong local RF carrier, or by unstable PLL clock-recovery.

Bounded uncorrelated jitter is commonly caused by crosstalk from nearby data line. Bounded uncorrelated jitter is also called crosstalk induced jitter, which is the term that will be used in this paper. Unlike periodic jitter, crosstalk induced jitter is non-periodic. Crosstalk induced jitter is bounded by the finite coupling strength. It is uncorrelated, because there is no correlation to the channel’s own data. Rather, it is correlated with the data on the adjacent traces. Crosstalk induced jitter is difficult to distinguish from random jitter, leading to increased estimates of both random jitter and total jitter [3], since the pdf for total jitter is a convolution of the pdf of each jitter component.

Several recent studies have focused on understanding, modeling, and measuring crosstalk induced jitter. A simple crosstalk induced jitter model was developed in [4] to calculate the time difference between the distortion-free and the distorted edge crossings of the victim signal. While this model was validated through H-spice simulation and measurement, the validation did not consider the effect of random and other types of jitter. Circuits to minimize the contribution of crosstalk induced jitter were proposed in [5] and [6]. The authors propose a model for the crosstalk induced jitter and then suggest

an equalizer circuit which compensates for the crosstalk, based on the coupling between the two circuits and the data on each. The authors of [7] present techniques to measure the amount of crosstalk induced jitter independent of other jitter sources. These techniques are based on an assumption that the crosstalk induced jitter dominates the total jitter in the system.

In the real word, crosstalk induced jitter is always associated with random and other types of jitter. Currently, there is no reliable method to decompose the total jitter into the different jitter components. Jitter decomposition methods are needed to verify the crosstalk induced jitter model. De-convolution can be used to decompose total jitter, but is very sensitive to noise [8]-[11], as will be illustrated in the following paper.

Methods are presented in the following paper to characterize deterministic and crosstalk induced jitter. The work is presented in five sections. A crosstalk induced jitter model is discussed in Section II which can be used to characterize the crosstalk induced jitter when the level of crosstalk is known. This model is validated through simulations and later through measurements. Methods for characterizing deterministic and crosstalk induced jitter are presented in Section III. A method is introduced in Section III.A for accurately characterizing random jitter in a measurement of total jitter. This characterization can then be used to determine the overall impact of deterministic jitter in the measurement. A method is presented in Section III.B for characterizing crosstalk induced jitter in measurements of total jitter made with and without the crosstalk sources present.

De-convolution of the pdf for crosstalk induced jitter is enabled using a Wiener filter. The models and methods are validated through measurements on a high-speed link

in Section IV. Results validate the crosstalk induced jitter model and demonstrate the effectiveness of the de-convolution method.

II. CROSSTALK INDUCED JITTER MODEL

A simple crosstalk model is illustrated in Fig. 2. When the circuits are electrically long, coupling along transmission lines must be taken into account [12]. Analytic equations for crosstalk are given in [13] for 6 different transmission line terminations. The maximum amplitude of crosstalk induced noise is given by [13], [4]

$$V_p = \frac{v_1 \cdot l \cdot \sqrt{L_{11} \cdot C_{11}}}{2 \cdot t_r} \cdot \left(\frac{L_{21}}{L_{11}} - \frac{C_{21}}{C_{11}} \right) \tag{1}$$

where v_1 is the voltage at the aggressor source, l is the length of the trace, t_r is the rise time of the aggressor signal, C11 and L11 are the per unit length self-capacitance and self-inductance of the victim, respectively, and C21 and L21 are the per unit length mutual capacitance- and mutual-inductance between the aggressor and victim. The maximum noise is important since it determines the peak-to-peak jitter.

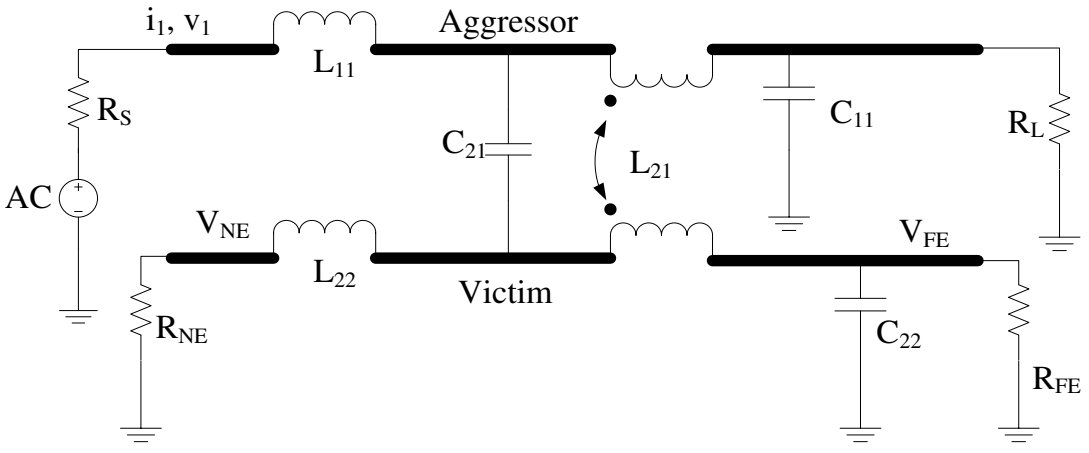

Fig. 2. Coupled circuit schematic for crosstalk noise analysis

A. Crosstalk Induced Jitter Model with Single Crosstalk Source

The jitter resulting from crosstalk can be determined from (1) and information about the victim signal. Figure 3 and Figure 4 shows an example of an aggressor and victim signal, and the resulting crosstalk induced jitter. Crosstalk causes the rising and falling edge of the victim signal to shift up or down in voltage, causing a corresponding shift in the time the edge occurs. The peak-to-peak crosstalk induced jitter is given by [4]:

$$t_{Xtalk} = t_{right} + t_{left} = \left|V_p\right| \cdot \frac{2t_v}{V_{victim}} \tag{2}$$

where t_{right} is the rightward shift in the edge of the victim signal, t_{left} is the leftward shift in the victim signal edge, V_{victim} is the full-range swing of the victim signal, and t_v is the rise time of the victim signal.

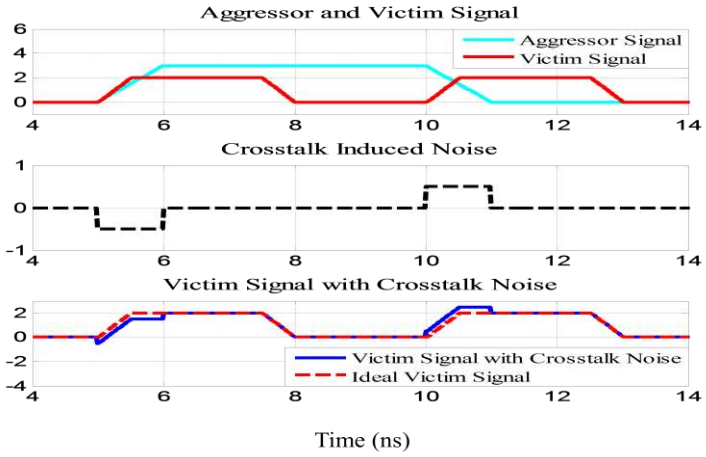

Fig. 3. Aggressor, victim, and crosstalk induced noise waveforms

Simulations were performed in Simulink to validate this model [14]. The time interval error and the crosstalk induced jitter histogram were calculated for a simple crosstalk problem with a single culprit, as in Fig. 2, and an aggressor and victim signal similar to Fig. 3, where the victim signal was twice the frequency of the aggressor. The

error between the simulation and the estimate from (2) was 4%, illustrating the accuracy of the model. This model can be used to estimate the peak-to-peak crosstalk induced jitter.

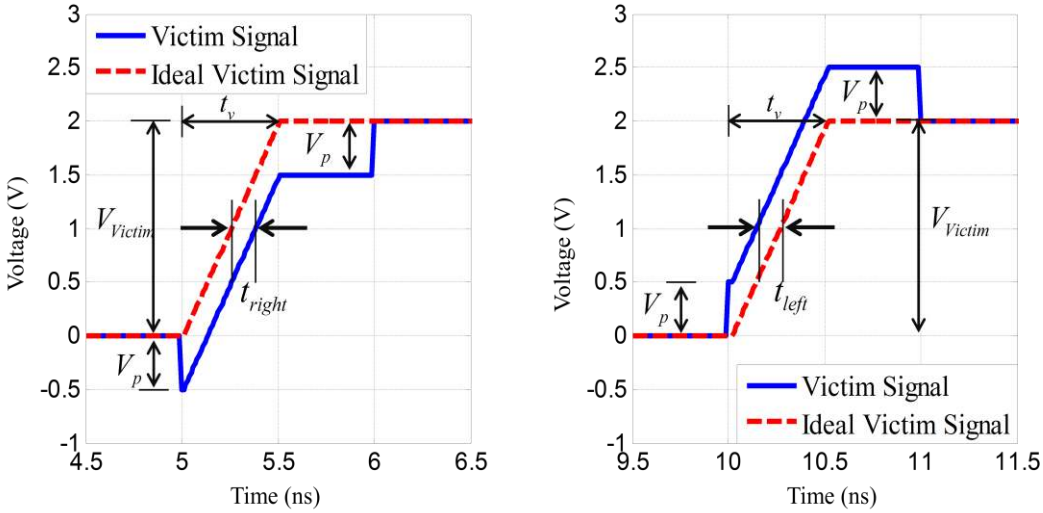

Fig. 4. Relationship between crosstalk induced jitter and magnitude of crosstalk induced noise

B. Crosstalk Induced Jitter Model with Multiple Crosstalk Sources

When multiple sources (traces) are present, the maximum crosstalk induced jitter can be approximated by assuming the crosstalk induced noise from each source is independent. For N aggressors, the peak-to-peak crosstalk induced jitter is given by [4]:

$$t_{Xtalk} = \left(\left| V_{p1} \right| + \left| V_{p2} \right| + \cdots + \left| V_{pN} \right| \right) \cdot \frac{2t_v}{V_{victim}} \tag{3}$$

where V_{pN} is the amplitude of crosstalk noise caused by the N_{th} aggressor.

Additional simulations were performed in Simulink to validate this model. The simulation schematic shown in Fig. 5 was built to model crosstalk within a 4-trace system. The 3 aggressors and victim are all at unique frequencies (i.e. not at harmonics of

one another), so are not in phase. The induced noise was up to 0.5 V, with positive and negative pulses similar to Fig. 3. The simulated waveform is shown in Fig. 6.

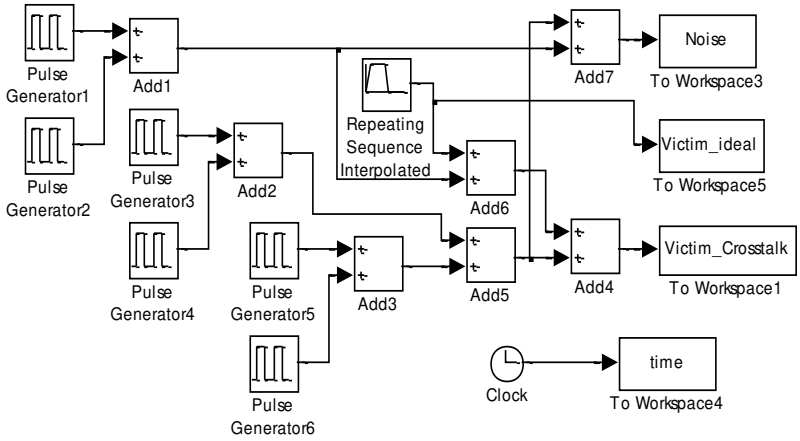

Fig. 5. Simulink schematic to simulate crosstalk induced jitter from multiple aggressors

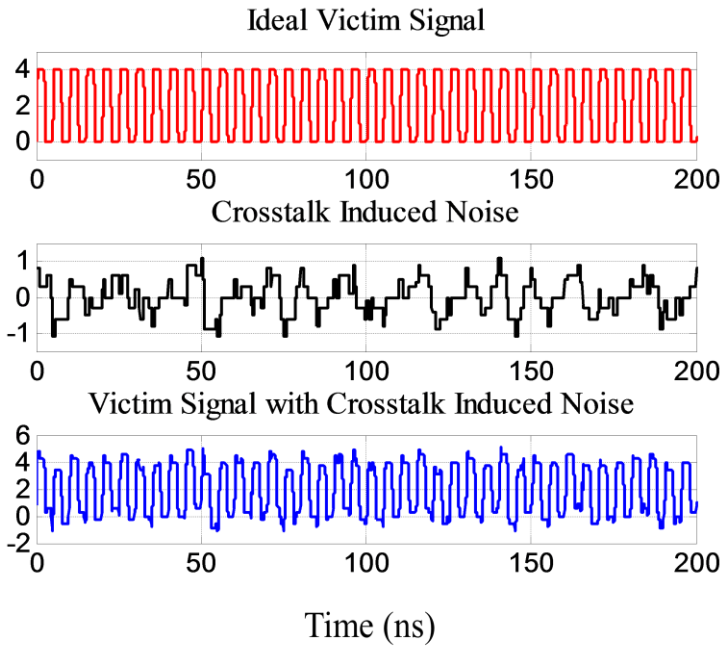

Fig. 6. Impact of crosstalk noise from multiple aggressors on victim signal

The time interval error and a histogram of crosstalk induced jitter were calculated from the simulated victim waveform, as shown in Fig. 7. An averaging function was used

to prevent "double counting" of edges (e.g. where a small amount of noise caused two transitions across Vdd/2 at an edge). Peak-to-peak crosstalk induced jitter was 0.4 ns in simulation compared to 0.35 ns predicted by the crosstalk induced jitter model. The use of averaging is expected to add some error to the simulation result.

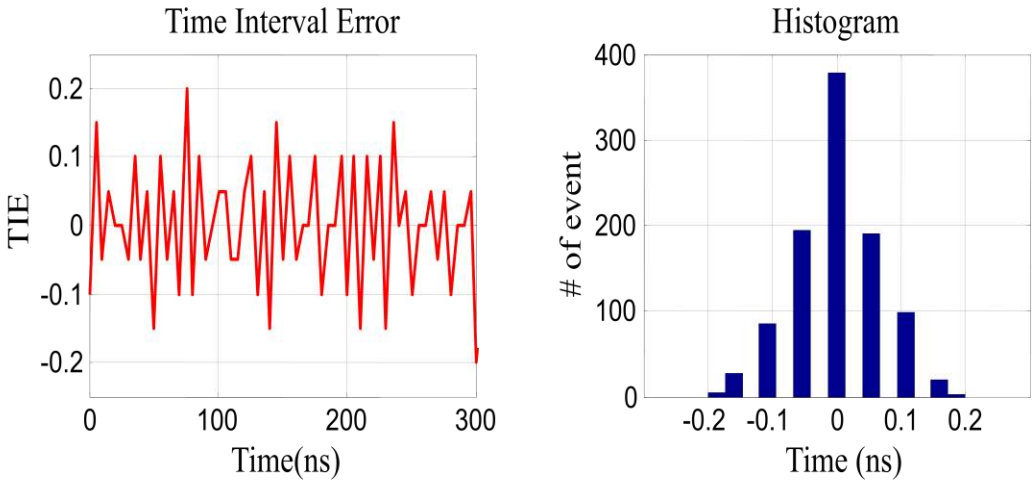

Fig. 7. Time interval error and histogram of jitter with crosstalk

III. DE-CONVOLUTION METHODS

The pdfs for deterministic jitter, crosstalk induced jitter, or random jitter cannot be measured independently. They are naturally mixed together in the measurement of total jitter. The individual components of jitter can be estimated, however, through tail-fit or de-convolution methods.

The pdf for total jitter is determined by the convolution of the pdfs for each jitter component [1]. For example, the pdf for total jitter is given by a convolution of the pdf of deterministic jitter and random jitter:

$$f_{TJ}(\Delta t) = f_{DJ}(\Delta t) * f_{RJ}(\Delta t) = \int_{-\infty}^{\infty} f_{DJ}(\tau) * f_{RJ}(\Delta t - \tau) d\tau \tag{4}$$

where f_{TJ} is the pdf for total jitter, and f_{DJ} and f_{RJ} are the pdfs for deterministic and random jitter, respectively. The pdf for deterministic jitter is similarly given by a convolution of the probability density for crosstalk induced jitter, periodic jitter, and data-dependent jitter.

A. Tail-fit De-convolution Method

Tail-fit methods can be used to separate total jitter into its random and deterministic components [14]-[20]. The pdf for random jitter is estimated from measurements of total jitter. Assuming the random jitter has a Gaussian distribution; measurements of a point at the "peak" of the distribution and at one point along the tail of the total jitter can be used to estimate the mean and standard deviation of random jitter. The difficulty with the standard tail-fit approaches is that the peak if often difficult to define and points along the tail are noisy. The method only works well for histograms with an "ideal" shape which is constructed from a large number of measurements. Resulting estimates of random noise are often insufficient for use with processes with very low bit error rates.

A new tail-fit method was designed to provide a better estimate of random jitter [14]. As with previous techniques, it was assumed that the tail of the total jitter histogram results from random jitter which has a Gaussian distribution. Instead of picking only two points to estimate the characteristics of random jitter, the entire tail is fit to the tail of a Gaussian distribution plot. Once the tail is found, one can determine where the curve diverges from the Gaussian distribution. This point of divergence determines the bounds for the deterministic jitter (i.e. it is the point where deterministic jitter dominates

the overall histogram for jitter). These bounds approximate the peak-to-peak deterministic jitter.

The shape of a Guassian distribution is determined by its mean μ and standard deviation σ. The parameters which best fit the tail can be found by sweeping the values of μ and σ from their minimum to maximum values while comparing the resulting curves to the measured histogram. The quality of fit can be quantified by the parameter Fit_threshold, defined as:

$$Fit\_th = \frac{\sum_{i=0}^{i=n}\left(\left|\frac{A(i)-B(i)}{A(i)}\right|+\left|\frac{A(i)-B(i)}{B(i)}\right|\right)}{n} \tag{5}$$

where A and B represent points on the calculated Gaussian distribution curve and measured histogram tail, respectively, and n is the number of points evaluated along the tail. The tail fit quality is better for smaller values of Fit_threshold.

The proposed tail-fit method was validated through measurements. A J-BERT high-performance serial BERT was used to generate a jitter source. The total jitter was composed mostly of periodic and random jitter. Fig. 8 shows the measured total jitter (in blue), the measured random jitter (in black), and the predicted random jitter (in red) found using the proposed tail fit method. The random jitter was measured directly from the random "source" within the J-BERT instrument. The proposed method did a good job of estimating the pdf for random jitter. The peak-to-peak deterministic jitter, which is dominated by peak-to-peak period jitter in this case, is the time-difference between the peaks of the two Gaussian distributions in Fig. 8. Here, the periodic jitter should span from -9.2 to 9.5 ps, which closely matches the predicted span.

The tail-fit de-convolution method works well if the system contains very limited crosstalk induced jitter, so the impact of crosstalk induced jitter on the tail is small. If the crosstalk induced jitter is large, the tail of the total jitter pdf contains both crosstalk induced jitter and also random jitter. In this case, this technique cannot independently find deterministic jitter.

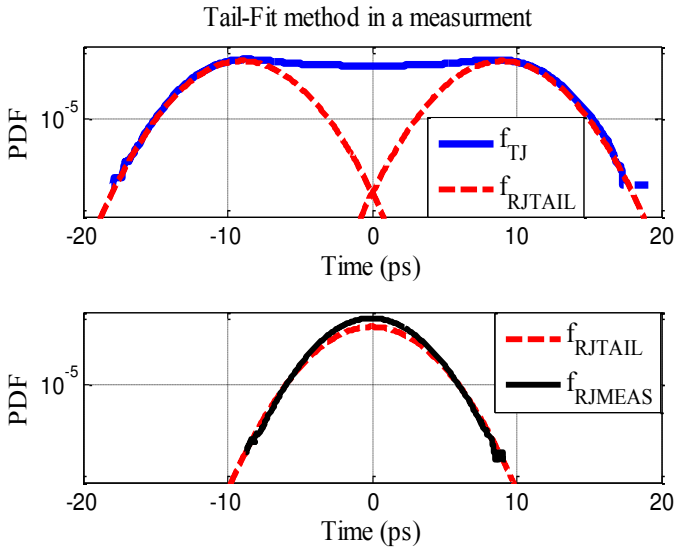

Fig. 8. Probability density functions for measured total jitter (blue), measured random jitter (black), and estimate random jitter (red)

B. Inverse Filtering De-convolution Method

The pdf for deterministic jitter can be found through de-convolution of the pdfs of measured total jitter and random jitter. Working in the frequency domain, the characteristic function for deterministic jitter (i.e. Fourier transform of the pdf) is given by:

$$F_{DJ}(s) = \frac{F_{TJ}(s)}{F_{RJ}(s)} \tag{6}$$

where $F_{DJ}(s)$, $F_{TJ}(s)$, and $F_{RJ}(s)$ are the characteristic functions for the estimated deterministic jitter, measured total jitter, and estimated random jitter, respectively [21]. The pdf for jitter can be determined by inverse Fourier transform. For example,

$$f_{DJ}(\Delta t) = \Im^{-1}\left\{F_{DJ}(s)\right\} \tag{7}$$

where $f_{DJ}(\Delta t)$ is the probability density and $\Im^{-1}\{\square\}$ is the inverse Fourier transform.

While the de-convolution approach in (7) may work reasonably well for random jitter, which has a well defined pdf, it does not work as well for crosstalk induced jitter. One possibility for using de-convolution to determine crosstalk induced jitter is through measurements of total jitter with and without the crosstalk sources present, as shown in:

$$f_{CJ}(\Delta t) = \Im^{-1}\left\{\frac{F_{TJ}(s)}{F_{NCJ}(s)}\right\} \tag{8}$$

where f_{CJ} is the predicted pdf for crosstalk induced jitter, F_{TJ} is the characteristic function for total jitter measured with crosstalk induced jitter, and F_{NCJ} is the characteristic function for total jitter without crosstalk induced jitter. F_{TJ} is found when all the crosstalk noise sources are turned on, and F_{NCJ} is found when the crosstalk noise sources are turned off.

This de-convolution method works well when the measurement noise is very small. Fig. 9 shows an example where (8) was used to estimate crosstalk induced jitter in a system with minimal noise. The top plot in Fig. 9 shows two histograms of total jitter that contain the same amount of crosstalk induced jitter, but different amounts of jitter from other sources. The second plot shows the total jitter found when the crosstalk sources are turned off. The bottom plot shows the estimated and actual crosstalk induced

jitter found using (8). The estimated and actual pdfs for crosstalk induced jitter match very well, but noise was nearly zero in this case.

Any practical measurement will contain non-negligible amounts of noise. This noise can corrupt the estimate of crosstalk induced jitter, because the inverse filter used for de-convolution is not stable [22]-[28]. This instability means that a small error in the measured total jitter can lead to large errors in the estimated crosstalk induced jitter. This effect is illustrated in Fig. 10. In the top plot, the pdf for crosstalk induced jitter was estimated using (8) when total jitter when the signal-to-noise ratio was 100. Even though the noise was only 1% of the measured pdf, the crosstalk induced jitter is unrecognizable. This problem can be solved in part by using a regularization technique, such as can be obtained using a Wiener filter.

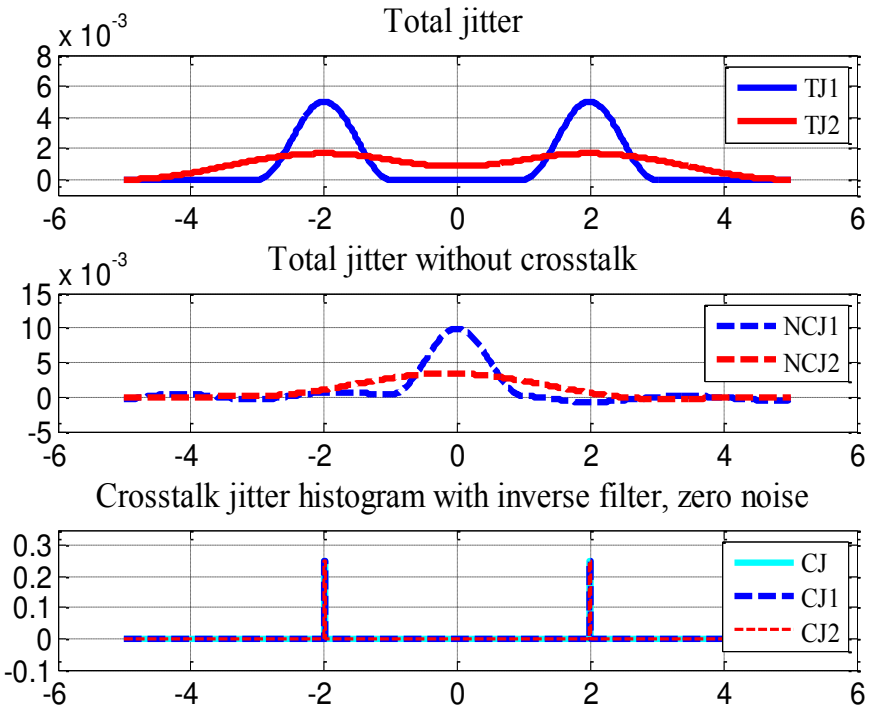

Fig. 9. Probability density functions for total jitter found with different amounts of non-crosstalk induced jitter (top), total jitter without crosstalk, actual and estimated crosstalk induced jitter

C. De-convolution Using a Wiener Filter

A Wiener filter is used to estimate the input to a linear system from a measurement of the output. The Wiener filter minimizes the mean square error in the estimate given information about the signal and noise power.

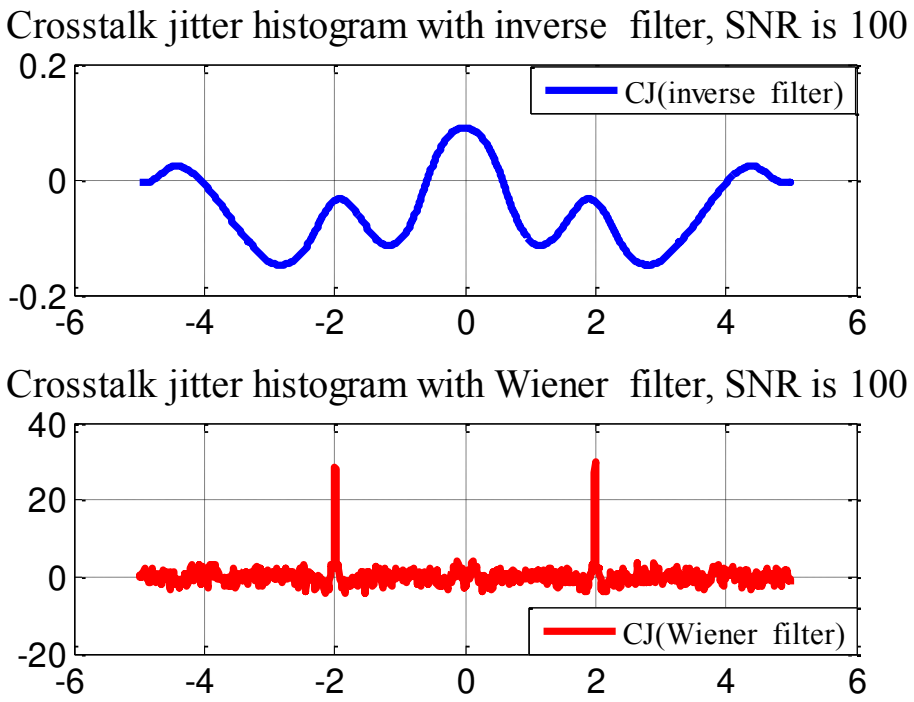

Fig. 10. Estimates of crosstalk induced jitter made in the presence of noise. Top: estimated using (8). Bottom: estimated using (15)

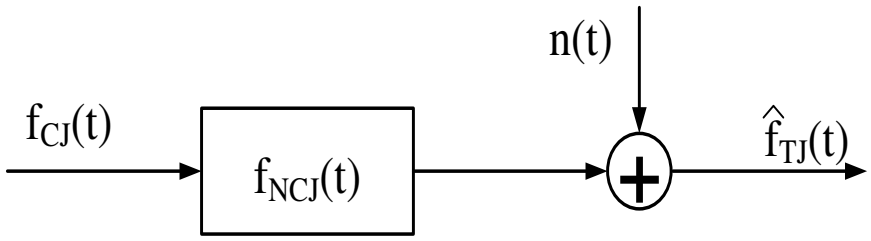

Fig. 11. Block diagram for the generation of a total jitter pdf measurement

The block diagram in Fig. 11 approximates the creation of the total jitter histogram in the presence of crosstalk and other source of jitter, as well as measurement noise, n(t). The measured histogram for the crosstalk induced jitter is given by:

$$\hat{f}_{TJ}(t) = f_{CJ}(t) * f_{NCJ}(t) + n(t) \tag{9}$$

where $\hat{f}_{TJ}(t)$ is the measured estimate of the pdf of total jitter. This relationship can be represented in the frequency domain as:

$$\hat{F}_{TJ}(\omega) = F_{CJ}(\omega) \cdot F_{NCJ}(\omega) + N(\omega) \tag{10}$$

The Wiener filter is used to eliminate the effect of measurement noise. For the system in Fig. 10, the Wiener filter is given by [18]:

$$W = \frac{F_{NCJ}{}^{*}(\omega) \cdot P_{TJ}}{\left|F_{NCJ}(\omega)\right|^2 \cdot P_{TJ} + P_N} \tag{11}$$

where P_{TJ} is power spectra of the total jitter histogram and P_N is power spectra of measurement noise. $F_{NCJ}{}^{*}(\omega)$ stands for the conjugate transpose of $F_{NCJ}(\omega)$. These power spectra are given by:

$$P_{TJ} = F_{TJ}{}^{*}(\omega) \cdot F_{TJ}(\omega) \tag{12}$$

$$P_N = N^{*}(\omega) \cdot N(\omega) \tag{13}$$

Re-arranging (13), the optimal Wiener filter is given by:

$$W = \frac{F_{NCJ}{}^{*}(\omega)}{\left|F_{NCJ}(\omega)\right|^2 + \frac{1}{SNR}} \tag{14}$$

where SNR is the signal-to-noise ratio of the measured histogram for total jitter histogram, is equal to P_{TJ} / P_N.

Given that $F_{NCJ}(\omega)$ must also be measured in the presence of noise, a reasonable estimate for the pdf of crosstalk induced jitter can be made using an estimate of the Wiener filter as:

$$\begin{aligned}\hat{F}_{CJ}(\omega) &= W \cdot \hat{F}_{TJ}(\omega) \\ &= \frac{\hat{F}_{NCJ}^{*}(\omega) \cdot \hat{F}_{TJ}(\omega)}{\left|\hat{F}_{NCJ}(\omega)\right|^2 + \frac{1}{SNR}}\end{aligned} \tag{15}$$

The SNR is a regularization parameter which can be estimated from knowledge of the measurement accuracy and from the measurement of total jitter. The utility of this approach is illustrated in Fig. 10, where the bottom plot shows the estimate of crosstalk induced jitter for the case in Fig. 9 with addd noise. The formulation in (15) was able to reconstruct the pdf of crosstalk induced jitter when (8) was not.

IV. CROSSTALK INDUCED JITTER MODEL VALIDATION IN MEASUREMENTS

Measurements of jitter were made on a test PCB to demonstrate the accuracy of the jitter prediction models. The PCB is shown in the Fig. 12. This PCB consists of 1 oz copper layers (approximately 1.3 mil thick) and an FR402 dielectric that is 59 mils thick. The manufacturer's data sheet lists the FR402 dielectric constant as 4.25 at 1 GHz and a loss tangent of 0.015 at 1 GHz. Measurements were made separately on the top pair of traces (pair 1) and the bottom pair (pair 2). One trace acted as an aggressor and one as the victim.

Fig. 12. Test PCB board

A. Characterization of the Crosstalk

The level of crosstalk was needed to analytically form an estimate of the crosstalk induced jitter. The level of crosstalk was determined experimentally. A 70 MHz clock was applied to one trace, while the magnitude of crosstalk was measured on the second trace. The far ends of each trace were terminated with 50 ohms. The magnitude of the aggressor clock was varied from 0 V to 5 V while measuring the signal on the victim using an oscilloscope. The measured crosstalk as a function of the aggressor clock voltage is shown in Fig. 13. This relationship was used in the next experiments to define the magnitude of the crosstalk noise leading to jitter. If this crosstalk noise can be measured in a real system, it can be used directly with (2) or (3) to estimate peak-to-peak crosstalk induced jitter.

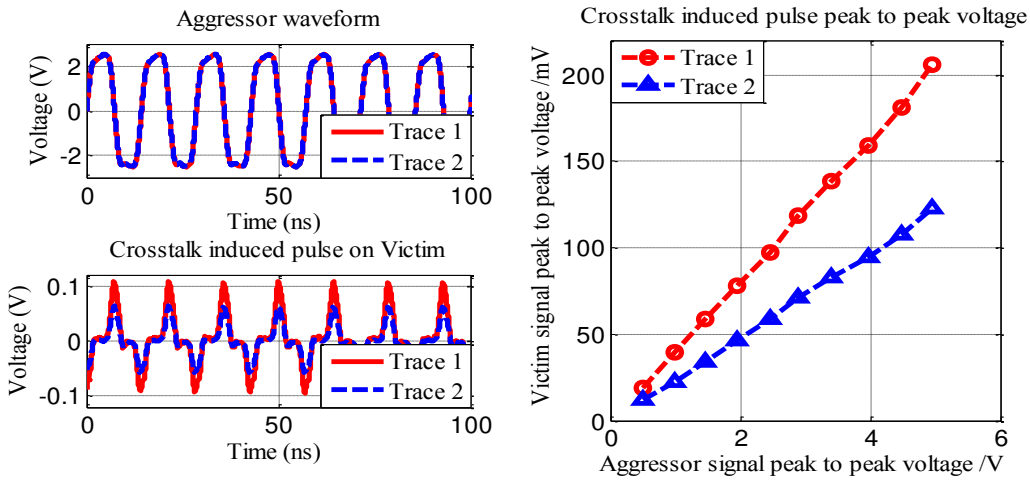

Fig. 13. Waveforms on aggressor and victim and the relationship between the magnitude of the aggressor clock and the magnitude of the crosstalk on victim

B. Crosstalk Induced Jitter Histogram

Histograms of the total jitter were measured with and without crosstalk to form a basis for estimating the crosstalk induced jitter. The measurement setup is shown in Fig.

14. The PRBS generator was connected to the victim trace and set to generate a 2 V, 400 MHz PRBS7 signal. Eye diagrams of the victim output signal were measured in the presence of crosstalk noise. The clock signal of the PRBS generator was used as the trigger signal to create the eye diagrams in the oscilloscope. A 70 MHz clock signal was applied to the aggressor trace while varying the magnitude of the clock signal. The magnitude of the resulting crosstalk was estimated from Fig. 13. While these signals are slow relative to signal speeds in modern systems, they are adequate for demonstrating the theory presented here.

Figure 15 shows the measured histogram of the total jitter in the victim output when there was no crosstalk in red (i.e. when the magnitude of the aggressor signal was zero and the jitter was strictly due to random jitter and data-dependent jitter) and when a 5 V clock signal was applied to the aggressor in black (resulting in crosstalk induced jitter as well). The estimated pdf for the crosstalk induced jitter, found using a Wiener filter (15), is shown in blue. The value of the signal to noise ratio was determined experimentally. The accuracy of the crosstalk induced jitter estimate will be demonstrated in the next section. Without a Wiener filter, a reasonable estimate cannot be made, as demonstrated in the bottom plot in Fig. 15. The result found using (8) is clearly unrealistic.

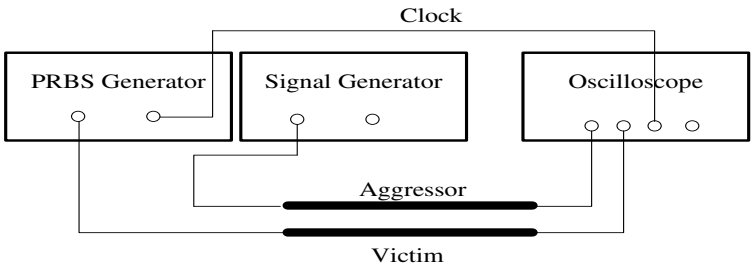

Fig. 14. Crosstalk induced jitter measurement setup

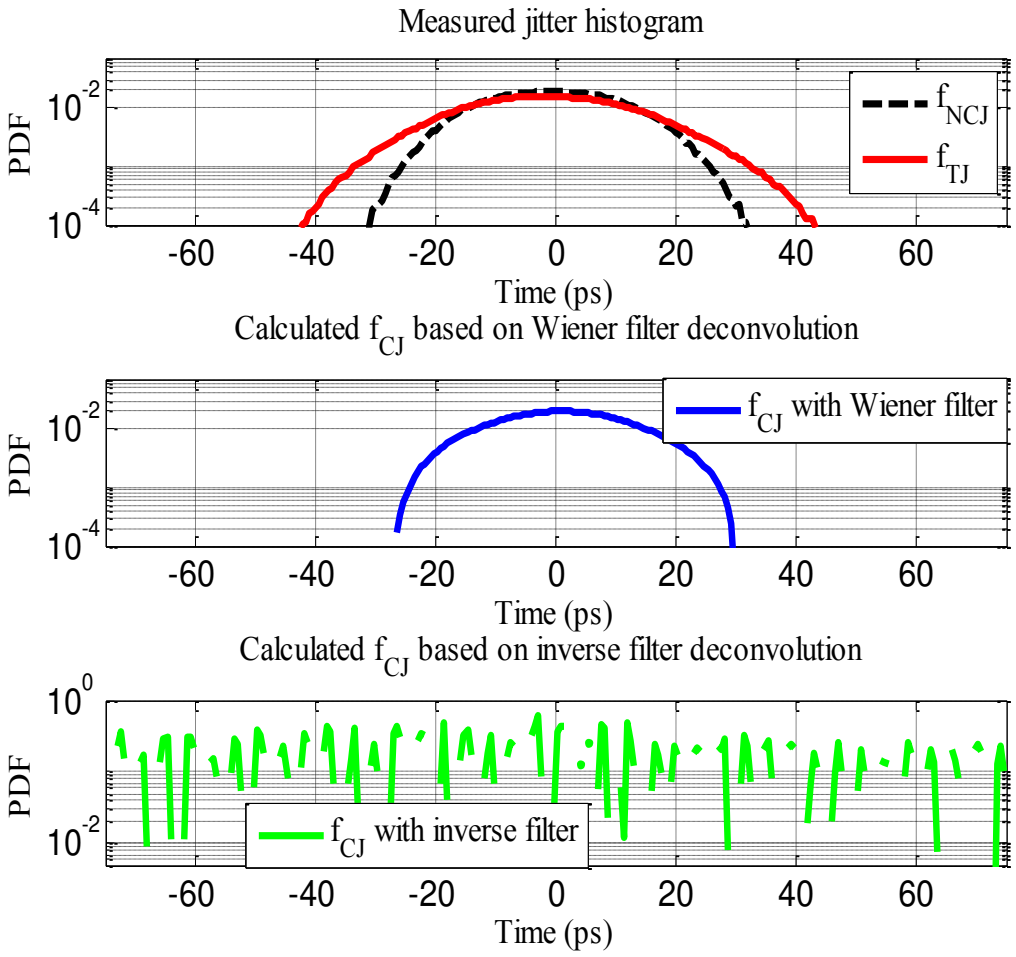

Fig. 15. Measured histogram for total jitter and estimated probability density function for crosstalk induced jitter using a Wiener filter (15) – middle – and using direct deconvolution as in (8) – bottom

Use of the crosstalk induced jitter model and of the method for estimating the pdf of crosstalk induced jitter was validated by estimating the peak-to-peak crosstalk induced jitter in both cases. The peak-to-peak jitter can be found from (2) or (3) when the value of crosstalk is known, or can be found from the maximum and minimum values of crosstalk in the crosstalk induced jitter pdf found using (15) (e.g. in in Fig. 14, the peak-to-peak jitter is roughly 29 ps – (–28 ps) = 57 ps). To validate these approaches, the peak-to-peak jitter was found using (2) and using (15) while varying the magnitude of the aggressor

voltage. The magnitude of the crosstalk used in (2) was found from Fig. 13. The resulting estimates of peak-to-peak crosstalk induced jitter are shown in Fig. 16. When the aggressor signal was 5 V, the crosstalk induced jitter accounted for roughly 50% of the total jitter. When the aggressor signal was 1 V, it accounted for roughly 20% of the total jitter. The estimated peak-to-peak jitter found using both (2) and (15) were nearly the same at all aggressor signal levels, suggesting both are valid methods of estimating the impact of crosstalk induced jitter.

At low aggressor voltages, there is a small mismatch in the estimated crosstalk induced jitter (roughly 10%). This mismatch is likely caused by mis-estimation of the jitter by the Wiener filter. The Wiener filter is applied to noisy measurements. While the Wiener filter is an optimal estimator in the presence of uncorrelated noise, it cannot completely eliminate the impact of that noise. Some errors must remain. The error can be reduced by increasing the SNR and by using a larger number of sample points in the measurement (in this case, a larger number of bins in the measured histograms).

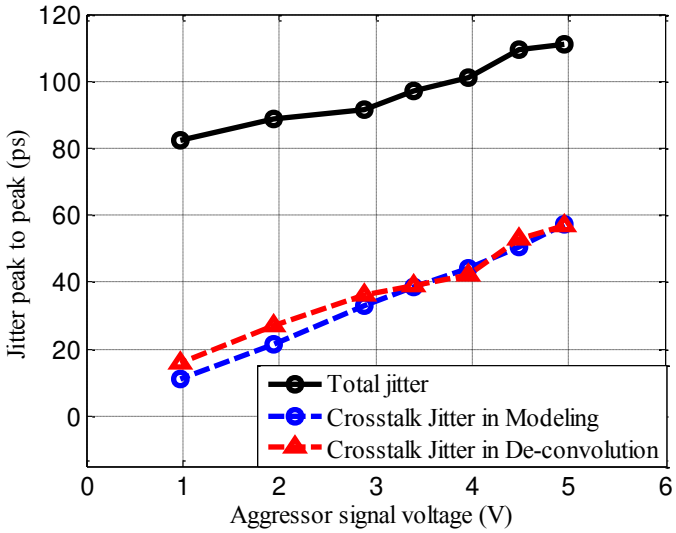

Fig. 16. Comparison of peak-to-peak crosstalk induced jitter in the model and in measurement

V. CONCLUSION AND FUTURE WORK

Crosstalk induced jitter can add significantly to the total jitter in high speed links, but determining the contribution of this jitter can be challenging. Methods were presented for determining the impact of crosstalk induced jitter analytically and from measurements. A method was proposed to estimate the pdf of random jitter from measurements of total jitter. This result can then be used to estimate the contribution of deterministic jitter (which includes crosstalk induced jitter). A method was also proposed for estimating crosstalk induced jitter from measurements of total jitter. The use of a Wiener filter allows the estimate to be made accurately even in the presence of measurement noise. The crosstalk induced jitter model and methods of estimating the contribution of crosstalk to jitter were validated both in simulation and in measurements with good results.

While a Wiener filter was shown to accurately estimate the pdf of crosstalk induced jitter, the technique may not work in all scenarios. One issue is that it may not be reasonable to make a measurement of jitter without crosstalk noise sources present. Such measurements are possible for special experimental measurements and test boards, but may not be reasonable for a fully working system. Another issue is that determining the level of SNR for use in (15) may not always be straightforward. Fortunately, even very rough estimates of SNR are typically sufficient for reasonable estimates of crosstalk induced jitter. For example in Fig. 16, nearly the same estimates of peak-to-peak crosstalk induced jitter are made using SNR estimates from 10-10000.

REFERENCES

[1] M. P. Li, Jitter, Noise and Signal Integrity at High-Speed, Prentice Hall Pearson Education, Chapter 1, pp.14-16, 2007.

[2] M. Marcu, S. Durbha, and S. Gupta, "Duty-cycle distortion and specifications for jitter test-signal generation." In Proc. IEEE Int. Symp. Electromagn. Compat, 2008.

[3] P. Zivny, M. Agoston, K. Engenhardt, and J. Carlson, "Method of BERanalysis of high-speed serial data transmission in presence of jitter and noise." presented at the DesignCon 2007, Santa Clara, CA, Jan. 2007.

[4] A. Kuo, R. Rosales, T. Farahmand, S. Tabatabaei, and A. Ivanov, "Crosstalk bounded uncorrelated jitter (BUJ) for high-speed interconnects." IEEE Trans. Instrumentation and Measurement, 54(5), 1800-1810, 2005.

[5] J. Buckwalter, and A. Hajimiri, "Cancellation of crosstalk-induced jitter." Solid-State Circuits, IEEE Journal of 41.3: 621-632, 2006.

[6] J. Buckwalter, and A. Hajimiri, "Crosstalk-induced jitter equalization." In Custom Integrated Circuits Conference IEEE, pp. 409-412, 2005.

[7] M. Miller, and M. Schnecker, "Quantifying Crosstalk-induced jitter in multi-lane serial data system." DesignCon, 2009.

[8] Q. Dou, J. A. Abraham, "Jitter decomposition in high-speed communication systems." In Test Symposium, 2008 13th European (pp. 157-162). IEEE. May, 2008.

[9] S. Erb, W. Pribyl, "Comparison of jitter decomposition methods for BER analysis of high-speed serial links." In Design and Diagnostics of Electronic Circuits and Systems, 2010 IEEE 13th International Symposium on (pp. 370-375). IEEE. April, 2010.

[10] C. d'Alessandro, V. Darsinos, B. Yegnanarayana, “Effectiveness of a periodic and aperiodic decomposition method for analysis of voice sources.” Speech and Audio Processing, IEEE Transactions on, 6(1), 12-23. Chicago, 1998.

[11] S. H. Hall, and H. L. Heck, Advanced Signal Integrity for High-Speed Digital Designs. John Wiley & Sons, 2011.

[12] D. B. Jarvis, “The effects of interconnections on high-speed logic circuits.” IEEE Trans. Electronic Computers, PP. 476-487, 1963.

[13] J. A. DeFalco, “Reflection and crosstalk in logic circuit interconnections.” Spectrum, IEEE, 7(7), 44-50, 1970.

[14] C. Sui, T. Zhu, C. Cheng, and D. Beetner, “A NewTail-Fit Method Design for Jitter Decomposition” Proc. IEEE Int. Symp. Electromagn. Compat,2014.

[15] N. Ou, T. Farahmand, A. Kuo, S. Tabatabaei, and A. Ivanov, “Jitter models for the design and test of Gbps-speed serial interconnects”. Design and Test of Computers, IEEE, 21(4), 302-313, 2004.

[16] B. Ham, “Fibre Channel-Methodologies for Jitter and Signal Quality Specification-MJSQ.” INCITS, Washington, DC 11, 2004.

[17] J. Patrin, and M. Li, “Comparison and correlation of signal integrity measurement techniques.” DesignCon, 2002.

[18] D. B. Jarvis, “The effects of interconnections on high-speed logic circuits,” IEEE Trans. Electron. Comput., vol. EC-12, no. 10, pp. 476–487, Oct. 1963.

[19] M. P. Li, J. Wilstrup, R. Jessen, and D. Petrich, “A new method for jitter decomposition through its distribution tail fitting.” International Test Conference, 1999.

[20] M. Shimanouchi, M. P. Li, and D. Chow, “New modeling methods for bounded gaussian jitter (BGJ)/noise (BGN) and their applications in jitter/noise estimation/testing.” International Test Conference, 2009.

[21] J. Tohka, and A. Reilhac, “A Monte Carlo study of deconvolution algorithms for partial volume correction in quantitative PET.” In Nuclear Science Symposium Conference Record, IEEE, Vol. 6, pp. 3339-3345, 2006.

[22] N. S. Nahman, and M. E. Guillaume, “Deconvolution of time domain waveforms in the presence of noise.” NASA STI/Recon Technical Report N, 82, 27682, 1981.

[23] R. Pintelon, P. Guillaume, Y. Rolain, J. Schoukens, and H. Van Hamme, “Parametric identification of transfer functions in the frequency domain-a survey.” IEEE Transactions on Automatic Control, 39(11), 2245-2260,1994.

[24] M. Van Noort, L. R. Van Der Voort, and M. G. Löfdahl, “Solar image restoration by use of multi-frame blind de-convolution with multiple objects and phase diversity.” Solar Physics, 228(1-2), 191-215, 2005.

[25] R. Keys, “Cubic convolution interpolation for digital image processing. Acoustics.” Speech and Signal Processing, IEEE Transactions on, 29(6), 1153-1160, 1981.

[26] J. I. Jackson, C. H. Meyer, D. G. Nishimura, and A. Macovski, “Selection of a convolution function for Fourier inversion using gridding computerised tomography application.” Medical Imaging, IEEE Transactions on, 10(3), 473-478, 1991.

[27] C. S. S. Burrus, and T. W. Parks, DFT/FFT and Convolution Algorithms: theory and Implementation. John Wiley & Sons, Inc, 1991.

[28] D. Kundur, and D. Hatzinakos, “A novel blind deconvolution scheme for image restoration using recursive filtering.” Signal Processing, IEEE Transactions on, 46(2), 375-390, 1998.

[29] V. Madisetti, Digital Signal Processing Fundamentals. CRC press. Section 3, 25-3, 2010.

II. Predicting Statistical Characteristics of Jitter Due to Simultaneous Switching Noise

Chunchun Sui, *Student Member, IEEE*, Liehui Ren, *Student Member, IEEE*, Xu Gao, *Student Member, IEEE*, Tun Li, *Student Member, IEEE*, James L. Drewniak, *Fellow, IEEE,* and Daryl G. Beetner, *Senior Member, IEEE*

***Abstract*—**Switching of logic gates is often responsible for significant power supply noise. Predicting the jitter resulting from power supply noise can be critical to analyzing the proper operation of high-speed devices. The statistical characteristics of jitter, such as the mean and standard deviation of jitter, can be used to place a meaningful bound on the worst-case timing margin and to estimate the bit error rate. While the statistical characteristics of the noise can be found through simulations of many input logic vectors, such simulations require significant computational effort and require methods for choosing suitable data vectors. Vectorless methods allow rapid analysis of switching without using predefined input data and can be used to understand which portions of the logic circuit contribute most to noise. In this paper, methods using vectorless techniques are presented to predict the mean and standard deviation of power supply noise on the printed circuit board (PCB), and the mean and standard deviation of the resulting peak-to-peak jitter in a driver on the same PCB. Predictions and measurements while using a relatively slow clock demonstrate the approach can determine the average and standard deviation of the peak power supply noise on the PCB and of the peak-to-peak jitter within less than 21%, which is sufficient for predicting how a specific logic design might impact jitter, and for proposing means to minimize that impact.

Index Terms—Integrated circuit noise, statistical analysis, power distribution, jitter, field programmable logic array. analytical models, noise measurement

I. INTRODUCTION

As the speed of integrated circuits (ICs) grows and timing margins shrink, signal integrity issues become increasingly important to IC design [1]. Power supply noise can have a major impact on signal integrity, as an increase in noise can result in an increase of jitter [2]. Switching of logic gates at clock edges is responsible for significant power supply noise. An ability to predict bounds on the noise and the resulting jitter could allow the engineer to account for the jitter and could be used to develop improved designs with less jitter and lower bit error rate. Predicting the jitter caused by the noise from a particular logic design, however, is challenging. It requires a method to estimate the reasonable worst-case switching current consumed by the logic, a method of predicting the noise voltage resulting from the dynamic current consumed by the logic, and a method for predicting the jitter from the resulting noise. While vector-based methods are available for predicting the power supply noise caused by switching in the logic [3], vector-based methods require substantial computational effort, since many vectors must typically be simulated [4]. Finding appropriate vectors is challenging, particularly early in the design process. The goal of this paper is to develop a method using vectorless techniques to predict the statistical characteristics of timing jitter due to simultaneous switching noise. The statistical characteristics can be used to place bounds on the expected jitter and improve the logic design.

Prediction of simultaneous switching noise has been a topic of several recent studies. Power supply noise can be found through simulations [3] or from closed-form expressions [5]. Power supply noise from logic implemented in a field programmable gate array (FPGA) was predicted in [3]. Prediction was enabled with a high-frequency model of

the power delivery network (PDN) of the die, package, and printed circuit board (PCB). The PCB was modeled using a cavity structure. Switching current was determined through simulations in Quartus II of the power consumed by the logic when responding to specific input data. This paper demonstrated that power supply noise can be modeled precisely when given sufficient information about the PDN and the IC switching currents. Similar results were found in [6]. Input data vectors, however, are not always available and simulating many random vectors is computationally expensive.

Vectorless methods are computationally efficient and do not require known input data vectors [7]-[9], [15]. Vectorless methods are typically used to predict the power consumed by a logic design. As such, most vectorless methods only find the average switching rate and the average current. In [9], methods were presented for determining the variance of switching and of current from clock-to-clock. Calculating both the mean and variance allows precise statistical bounds to be place on peak current or power consumption. These techniques will be used here to statistically characterize the power supply noise.

The relationship between power supply noise and jitter is well known. A method to correlate simultaneous switching noise with signal jitter was presented in [10]. This work studied how the PDN impedance affects signal jitter and voltage margin. A similar study in [11] showed that output jitter peaks when the simultaneous switching noise is at the resonance frequency or at half the resonance frequency. Jitter in the clock and the I/O of an FPGA was studied in [12] and [13]. These papers investigated the relationship between jitter and the number of switching registers, the frequency of the switching circuit compared to the clock, the relative location of the switching circuit and the clock, and the

on-die decoupling capacitance. Since these papers are based on measured results, the methods cannot be used in the pre-design stage. A method for predicting clock jitter was introduced in [14] based on the amount of charge consumed per clock cycle by the logic design. This paper demonstrated that the charge consumed per clock could do a reasonably good job of predicting jitter, independent of the implemented logic pattern.

Methods are presented in the following paper to predict the statistical characteristics of jitter due to simultaneous switching noise in FPGAs when the clock speed is relatively low and the noise is on the PCB. Results demonstrate the accuracy of the approach and lay the foundations for future studies with faster clock speeds and with logic and I/O sharing the same power supply on-die. The work is presented in six sections. A vectorless method for predicting the statistical characteristics of the switching current is presented in Section II. These results are used in Section III to predict the statistical characteristics of power supply noise on a PCB. Methods to predict the statistical characteristics of peak jitter due to switching noise are presented in Section IV. Prediction of jitter characteristics is validated in Section V through simulations and measurements. Conclusions are summarized in Section VI.

II. PREDICTION OF THE STATISTICAL CHARACTERISTICS OF SWITCHING AND CHARGE CONSUMED PER CLOCK

The statistical characteristics of switching in a logic circuit can be approximated directly from the logic when the statistical characteristics of the input data are known. The theory behind vectorless estimation is explained in detail in [9] and [15]. The results are summarized below.

A. Mean and Standard Deviation of Switching

Consider the logic circuit in Fig. 1. The average number of switching events per clock at output y can be calculated as [15]:

$$\mu_y = \sum_{j=1}^{n} \mu_{y \leftarrow x_j} = \sum_{j=1}^{n} P\left(\frac{\partial y}{\partial x_j}\right) \mu_{x_j} \tag{1}$$

where μ_y is the average number of transitions per clock on *y*, $\mu_{y \leftarrow x_j}$ is the average number of transitions on *y* caused by input x_i, $P\left(\frac{\partial y}{\partial x_j}\right)$ is the probability of a transition at *y* due to a transition at x_j, μ_{x_j} is the average number of transitions on input *i*, and *n* is the total number of inputs.

The average number of switching events per clock over many logic gates is given by a sum of the average number of switching events occurring at each gate:

$$\mu_{total} = \sum_{i=1}^{N} \mu_i \tag{2}$$

where μ_{total} is the average number of switching events over all gates and μ_i is the average number of switching events at the output of logical gate *i.*

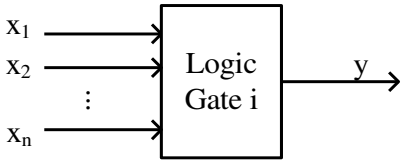

Fig. 1. A generic logic gate with n-inputs and one output

The standard deviation in the number of switching events at output y among clock cycles is given by [9]

$$\sigma_y^2 = \sum_{i=1}^{n}\left\{P^2\left(\frac{\partial y}{\partial x_i}\right)\sigma_{x_i}^2 + P\left(\frac{\partial y}{\partial x_i}\right)\left[1-P\left(\frac{\partial y}{\partial x_i}\right)\right]\left(\mu_{x_i}^2+\sigma_{x_i}^2\right)\right\} + \sum_{i=1}^{n}\sum_{j=1(i\neq j)}^{n} COV(x_i,x_j)P\left(\frac{\partial y}{\partial x_i}\right)P\left(\frac{\partial y}{\partial x_j}\right) \tag{3}$$

where σ_y and σ_{x_i} are the standard deviation in the number of switching events among clocks at the output *y* and input x_i, respectively. $COV(x_i,x_j)$ is the covariance in the number of switching events per clock at gates x_i and x_j, and is given by

$$COV\left(x_i,x_j\right) = \sum_i\sum_j COV\left(A_i,B_j\right)\cdot P\left(\frac{\partial I}{\partial A_i}\right)\cdot P\left(\frac{\partial J}{\partial B_j}\right) \tag{4}$$

where A_i and B_j represent switching at the inputs to the logic gates with outputs x_i and x_j, respectively. The variance in the total number of switching events per clock over many logic gates can be found as [9]

$$\sigma_{total} = \sqrt{\sum_{i=1}^{n}\sigma_i^2 + \sum_{i=1}^{n}\sum_{\substack{j=1\\ i\neq j}}^{n} COV(y_i,y_j)} \tag{5}$$

where σ_i^2 is the variance in the number of switching events at the output of gate *i*, and $COV(y_i,y_j)$ is the covariance among switching at output nodes y_i and y_j.

B. Mean and Standard Deviation of Charge Consumed per Clock

The statistical characteristics of the charge consumed per clock cycle can be determined from the charge consumed per switching event. The average charge per clock consumed across the logic circuit is given by

$$\mu_{\text{Charge}} = \sum_{i=1}^{N}\mu_i Q_i \tag{6}$$

where μ_{Charge} is the average total charge consumed per clock, and Q_i is the charge consumed by each transition of output *i*. The standard deviation in the charge-per-clock is similarly given by [9]

$$\sigma_{\text{Charge}} \approx \sqrt{\sum_i Q_i^2 \sigma_i^2 + \sum_i \sum_j Q_i Q_j COV(y_i, y_j)} \tag{7}$$

where σ_{Charge} is the standard deviation in the charge consumed per clock.

III. VECTORLESS ESTIMATION OF POWER SUPPLY NOISE

Previous work [3] has shown that the power supply noise can be accurately predicted when the input current waveform is known, but it not clear that knowledge strictly of the statistical characteristics of charge consumption is sufficient. Statistical estimates of charge consumption only give equivalent information about the average current drawn over each clock cycle, not the current waveform. Before estimating the statistical characteristics of power supply noise, a study was performed to demonstrate that knowledge of the charge consumed per clock was sufficient to estimate power supply noise. Sufficiency was demonstrated by simulating power supply noise for a variety of input current waveforms which each consume the same total charge.

A. Power Delivery Network Impedance Model

To determine the impact of the switching current waveform on power supply noise requires an accurate model of the power delivery network impedance. The relationship between the switching current, I(f), and power supply noise voltage, V(f), is given by [3],[20]:

$$V(f) = Z(f) \cdot I(f) \tag{8}$$

where Z(f) is the transfer impedance of the power delivery network in the frequency domain, between the noise source (i.e. where I(f) is located) to the noise location (i.e. where V(f) is measured).

The FPGA and PCB studied here are shown in Fig. 2. The FPGA power supply is connected to the PCB at Port 2. All switching currents are generated at this port. Ports 1 and 3 are at other locations on the PCB, between the power and return planes.

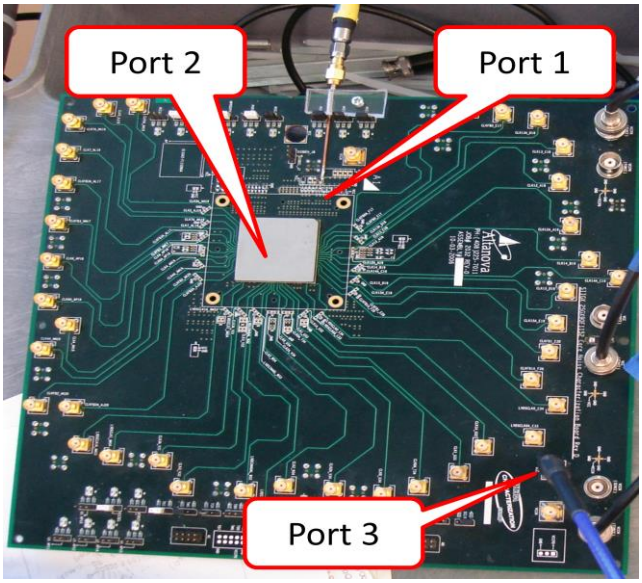

Fig. 2. Test PCB picture

A model of the impedance of the complete PDN, including the FPGA and PCB, was developed in [3]. A simplified model is shown in Fig. 3. The impedance of the PCB power and return plane are modeled using a cavity model, calculated using EZPP [21]. EZPP provides an S-parameter block of the power plane impedance. Measurements of the transfer impedance between ports in [3] showed that the simulated and measured impedances matched within a few dB from 100 kHz to 2 GHz. When current waveforms resulting from switching were predicted in [3] using Quartus, this model was shown to provide an accurate estimate of the PCB power supply noise.

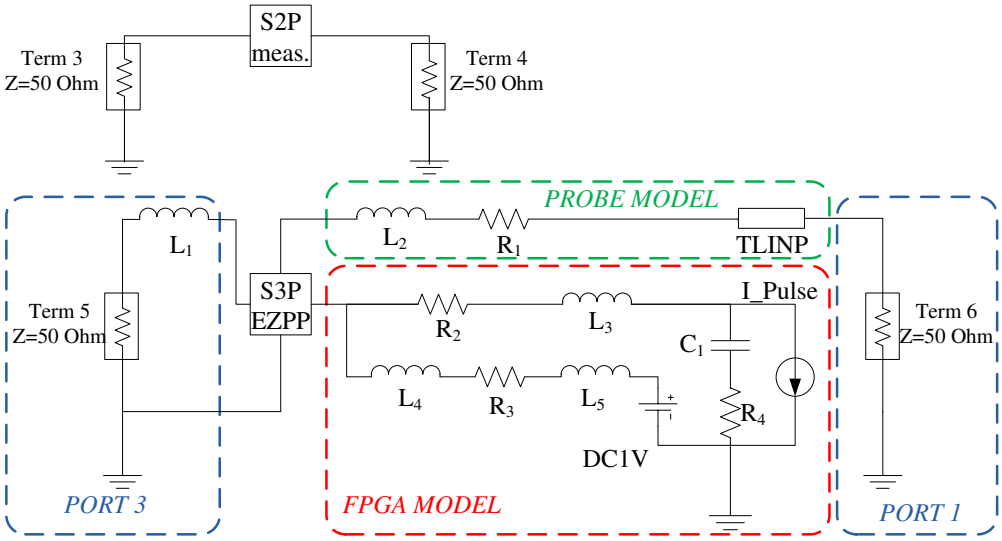

Fig. 3. Model of the die-package-PCB power delivery network

B. Switching Current Waveform

To demonstrate whether charge consumed per clock alone could be used to predict power supply noise, or if a switching current waveform was required, simulations of the power supply noise were made using multiple current waveforms consuming the same charge. Figure 4a shows three switching current waveforms in the FPGA, made assuming that switching occurs over a window of less than 8 ns (that smallest allowed clock period for the designs studied later). Figure 4b shows the power supply voltage at port 3 on the PCB during each noise waveform. The results demonstrate that each waveform, consuming the same charge per clock, generates roughly the same peak PDN noise (within about 10%). The peak noise voltage is consistent with charge until the pulse width is more than about 20 ns [22]. In the real design, all switching would end much earlier since the clock period can be as small as 8 ns.

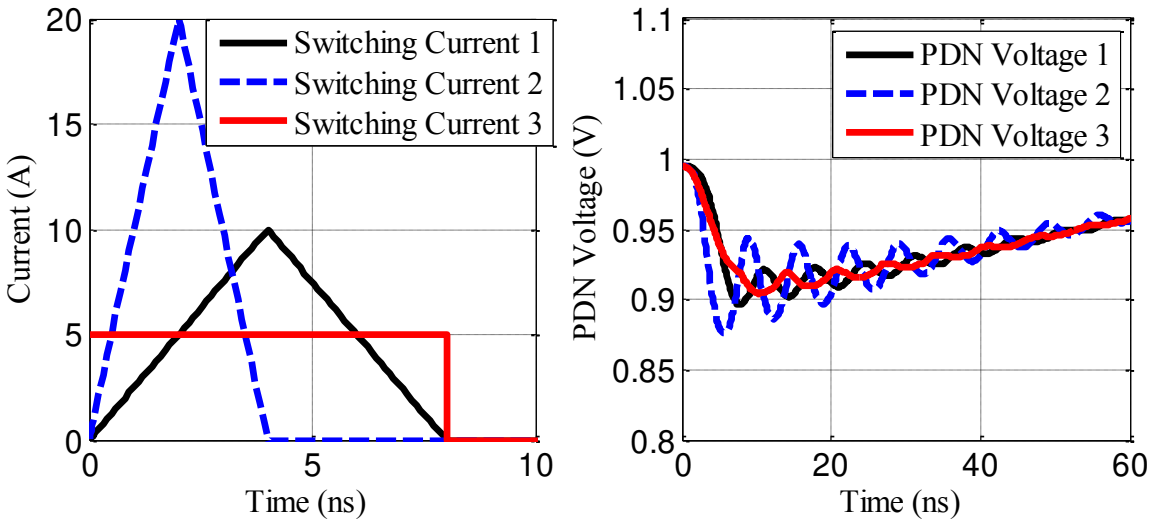

Fig. 4. Simulated power supply noise with different noise current waveforms which have the same average charge per clock

Simulations in Quartus provide information about power rather than current or charge consumption. The charge consumed per clock is roughly proportional to the average power over the clock when the magnitude of the power supply voltage is relatively small. That is,

$$P_{AVG,i} \approx V_{DD} \cdot I_{AVG,i} = V_{DD} \cdot \frac{Q_i}{T} \tag{9}$$

where $P_{AVG,i}$ and $I_{AVG,i}$ are the average power and current consumed over clock cycle i, Q_i is the consumed charge, and T is the clock period. The voltage drop in Fig. 4b is directly proportional to the total charge consumed, so

$$V_{DROP} \approx K \cdot P_{AVG} . \tag{10}$$

$$K \approx \frac{V_{DROP}}{V_{DD} \cdot Q_i} \cdot T \tag{11}$$

From Fig. 4, the value of K was found to be 1.22 mV/mW. This value of K will be used in the next section to estimate the voltage drop on the PCB resulting from a power per clock estimate from Quartus II. A comparison between the measured and predicted voltage

drops will be made to validate the model of the switching current and the power delivery network. Once validated, the value of K will be used with vectorless predictions of charge to characterize the statistical nature of the PCB noise.

C. Validation of Power Delivery Network Model

To validate the prediction of noise characteristics, 1000 5-bit multipliers were implemented in the Altera FPGA pictured in Fig. 2. The multipliers were driven by a 7-bit pseudorandom bit-sequence (PRBS7) generated by a linear feedback shift register. Fig. 5a shows the connection of the multipliers and the linear feedback registers. The resulting power supply noise at port 3 on the PCB was measured using the setup shown in Fig. 5b.

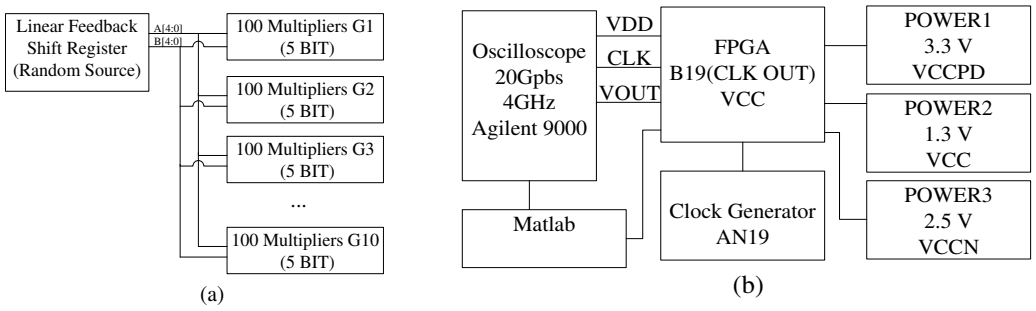

Fig. 5. (a) Logic circuits in FPGA; (b) PDN noise measurement setup

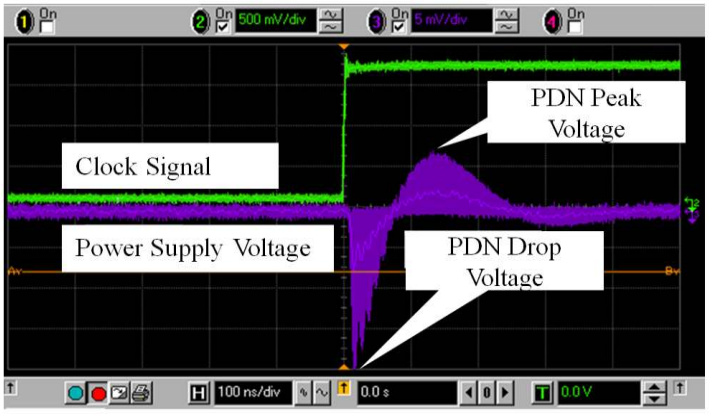

Fig. 6. Measured power supply noise on rising edges of the clock. The plot shows the measurement over many clocks, effectively showing the envelope of the noise

The average power consumed for each clock cycle was also predicted through simulations with Quartus II Power Play. The switching power for each clock cycle was determined by the power consumed within 500 ns of each rising edge of the clock. The voltage drop on the PDN was estimated by multiplying the simulated switching power by K, as in (10).

An example of the measured noise is shown in Fig. 6. The power supply voltage first drops, when the gates switch, then rises due to ringing. The noise was intentionally made large by removing most decoupling capacitors from the PCB. A comparison of the measured and predicted values of the maximum voltage drop is shown in Fig. 7 as a function of the clock number. The drop changes as the inputs (and the resulting noise) will vary from clock to clock. The measured and estimated drops match within an RMS error of 26 mV (26%), validating the simulation model and the method of estimating noise.

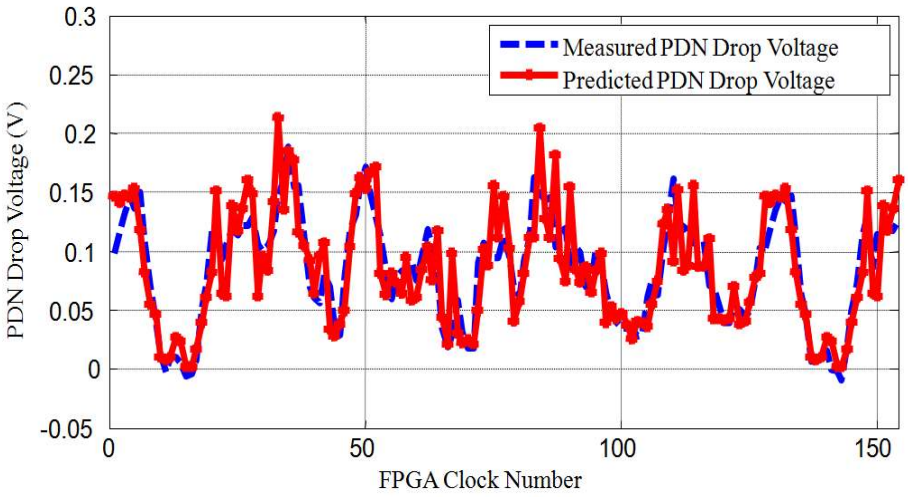

Fig. 7. Comparison of maximum voltage drop found in measurement and by multiplying simulated estimates of power with proportionality constant found from simulation

Since the maximum voltage drop is proportional to the consumed charge, the mean and standard deviation of the noise will be proportional to the mean and standard deviation of the consumed charge. That is,

$$\mu_{noise} \approx \frac{K \cdot V_{DD}}{T} \cdot \mu_{\text{Charge}} \tag{12}$$

$$\sigma_{noise} \approx \frac{K \cdot V_{DD}}{T} \cdot \sigma_{\text{Charge}} \tag{13}$$

A comparison of the statistical characteristics for the measured power supply noise, the noise predicted from the simulated power, and the noise predicted using (12) and (13) is shown in Table. 1. Comparisons between measured mean and standard deviation of the maximum voltage drop per clock and results predicted from simulated power are close, as expected. Results predicted using vectorless methods were within 2% of the experimental values for the mean and 8% for the standard deviation. Errors in the standard deviation may have been larger because of random noise in the measurement.

TABLE I

STATISTICAL CHARACTERISTICS OF POWER SUPPLY NOISE

Maximum Noise Voltage Drop on PCB PDN	Mean (ns)	Standard Deviation (ns)	Calculation time
Quartus simulation	86.9	47.9	12.7 hours
Measurement	86.8	43.1	1~3 hours
Vectorless prediction	93.6	44.0	5 minutes

IV. ESTIMATION OF PEAK JITTER

Power supply noise causes a change in the delay through an I/O or clock driver. The relationship between the noise and the resulting delay – or jitter – can be found with a driver model. An analytic model for delay is presented in the following section. This model will be used to estimate the jitter resulting from power supply noise, and later combined with results from the last section to provide vectorless estimates of the statistical characteristics of jitter resulting from logic switching.

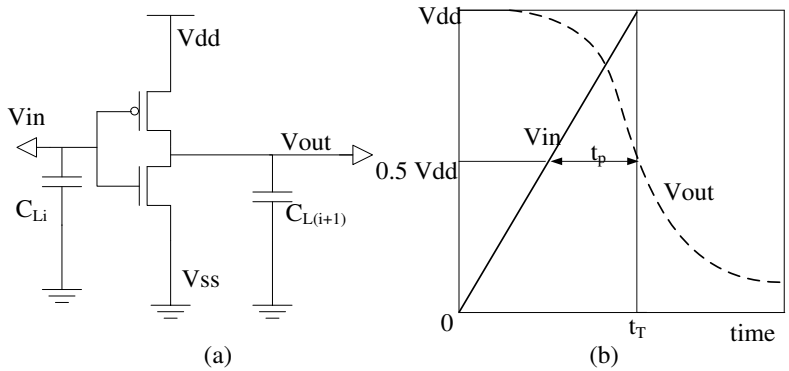

Fig. 8. Inverter delay model, (a) Inverter schematic, (b) Time domain waveform

A. Delay Model

The propagation delay through a generic digital logic gate is approximately [17]:

$$t_p = \frac{C_{Li}V_{dd}}{2I_{D0}} + (\frac{1}{2} - \frac{1-v_T}{1+\alpha})(\frac{0.9}{0.8} + \frac{V_{D0}}{0.8V_{dd}} In \frac{10V_{D0}}{eV_{dd}}) \frac{C_{L(i-1)}V_{dd}}{I_{D0}} \quad (14)$$

where V_{D0} is the drain saturation voltage at $V_{GS} = V_{dd}$, given by:

$$V_{D0} = (\frac{V_{dd} - V_{th}}{V_{dd,ref} - V_{th}})^{\alpha/2} V_{D0,ref}, \quad (15)$$

$V_{D0.ref}$ is the drain saturation voltage at $V_{GS} = V_{dd,ref}$, V_{dd} is the power supply voltage, V_{th} is the

MOSFET threshold voltage, α is the velocity saturation index for the MOSFET (typically from 1 to 2), ν_T is V_{th}/V_{dd}, and C_{Li} and $C_{L(i+1)}$ are the input capacitance and output capacitance driven by the gate, respectively. The propagation delay, t_p, is defined as the time between the input signal reaching half of V_{dd} to the output signal reaching half of V_{dd}, as shown in Fig. 8.

In a real application using an off-the-shelf device, the parameters in (14) and (15) may not be known to the user. A suitable model, however, can be created by rewriting the delay model in (14) as [17]

$$t_p = f(V_{dd}) = A \cdot g(V_{dd}) + B \cdot h(V_{dd}) \tag{16}$$

where $g(V_{dd})$ and $h(V_{dd})$ are given by:

$$g(V_{dd}) = \frac{0.9}{0.8}\left(\frac{1}{2} - \frac{1-\nu_T}{1+\alpha}\right) + \left(\frac{1}{2} - \frac{1-\nu_T}{1+\alpha}\right)\left(\frac{V_{dd} - V_{th}}{V_{dd,ref} - V_{th}}\right)^{\alpha/2} \frac{V_{D0,ref}}{0.8V_{dd}}\left(\frac{\alpha}{2} In\frac{V_{dd} - V_{th}}{V_{dd,ref} - V_{th}} + In\frac{10V_{D0,ref}}{eV_{dd}}\right) \tag{17}$$

$$h(V_{dd}) = \frac{V_{dd}}{(V_{dd,ref} - V_{th})^{\alpha/2}} \tag{18}$$

and

$$A = \sum_{i=1}^{M} \frac{C_{Li}(V_{dd,ref} - V_{th})^{\alpha}}{I_{D0,ref}} \tag{19}$$

$$B = \frac{1}{2}\sum_{i=2}^{M+1} \frac{C_{Li}(V_{dd,ref} - V_{th})^{\alpha}}{I_{D0,ref}} \quad . \tag{20}$$

A and B are constants which depend on the sizes FETs and capacitances in the logic circuit. The detailed derivation can be found in [17]. Since A and B are independent of the power supply voltage, they can be determined *a posteriori* by measuring the delay through

the gates for two different values of V_{dd}. Once known, these two constants can be used to predict the delay for other values of V_{dd}. These equations still require the parameters α, V_{th} and $V_{D0.ref}$. The selection of the velocity saturation index α is discussed in [18], which indicates the results are not sensitive to the selection of α. The parameters V_{th} and $V_{D0.ref}$ can be extracted from simulation. Using these parameters, (16) can be used to estimate jitter from the power supply noise.

B. Prediction of Peak Jitter

When the power supply voltage drops, as shown in Fig. 6, the delay through an I/O or clock driver will get longer, causing positive jitter with respect to the ideal switching point (See Fig. 9). When the power supply voltage peaks the delay will get shorter, causing negative jitter. The peak positive jitter, J_P, will occur at the maximum drop in voltage, V_{drop}, and the peak negative jitter, J_N, will occur at the maximum rise in voltage, V_{peak}. Since it is this maximum jitter that is most important, the following work will focus on finding the maximum jitter over a single clock cycle, and predicting the statistical characteristics of this maximum jitter as it varies from clock-to-clock.

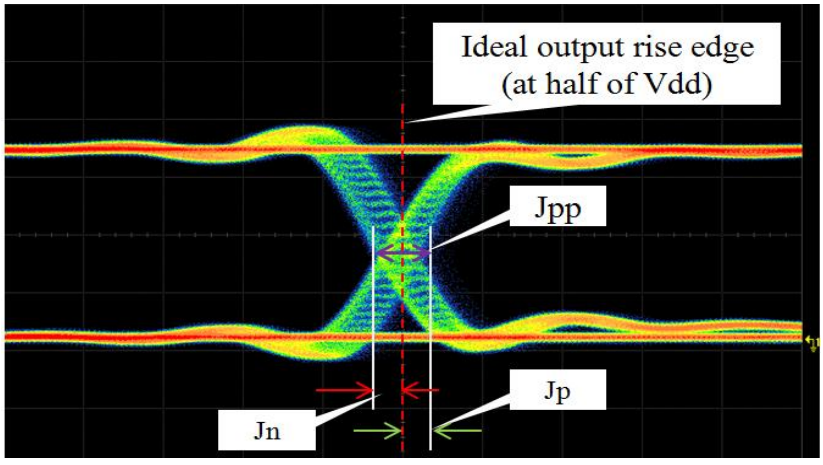

Fig. 9. Jitter diagram due to PDN noise

The peak positive and negative jitter over a clock cycle can be found from:

$$J_P = f\left(V_{DROP}\right) - f\left(V_{\mathrm{dd0}}\right) \tag{21}$$

$$J_N = f\left(V_{PEAK}\right) - f\left(V_{\mathrm{dd0}}\right) \tag{22}$$

where V_{dd0} is the typical power supply voltage, and $f(\square)$ is the function in (16) predicting the propagation delay through the driver at a particular supply voltage. The maximum peak-to-peak jitter from a single switching event is given by:

$$J_{PP} = J_P - J_N = f\left(V_{DROP}\right) - f\left(V_{PEAK}\right) \tag{23}$$

As shown in (23), the peak-to-peak jitter is determined from the maximum drop and peak in voltage due to noise. The statistical characteristics of peak jitter can be found from $f(\square)$ and the statistical characteristics of V_{drop} and V_{peak}. V_{drop} and V_{peak} are proportional to one another.

C. Model Validation

Models for delay and jitter were first validated by comparing jitter predicted from the measured power supply noise to the measured jitter. A high-speed inverter (Toshiba TC7SO4F) was selected as an example clock or I/O driver. The inverter power supply was connected through a capacitor to the PCB power supply at port 3, shown in Figs. 2 and 5. In this way, the power supply noise generated by the FPGA influenced the operation of the inverter. The FPGA was configured with 1000 5-bit multipliers driven by a 7-bit pseudorandom bit-sequence, as in section IIIB. The inverter and the FPGA were driven by different clocks, as would be common for many ICs on the same PCB.

The inverter was connected to the PCB as shown in Fig. 10a. The inverter supply was connected to the PCB through a DC decoupling capacitor, so the inverter could run on a 2.5 V DC, while the FPGA's supply voltage was much lower.

To correctly predict jitter requires the on-die noise voltage in the inverter, not just the noise on the PCB. The inverter package model is shows in Fig. 10(b). The package inductance (6.6 nH) and on-die decoupling capacitance (0.33 nF) were extracted from an impedance measurement. Simulation results showed that the on-die noise voltage was within 3% of the noise voltage on the PCB, so the PCB noise voltage was used directly when predicting jitter in later simulations.

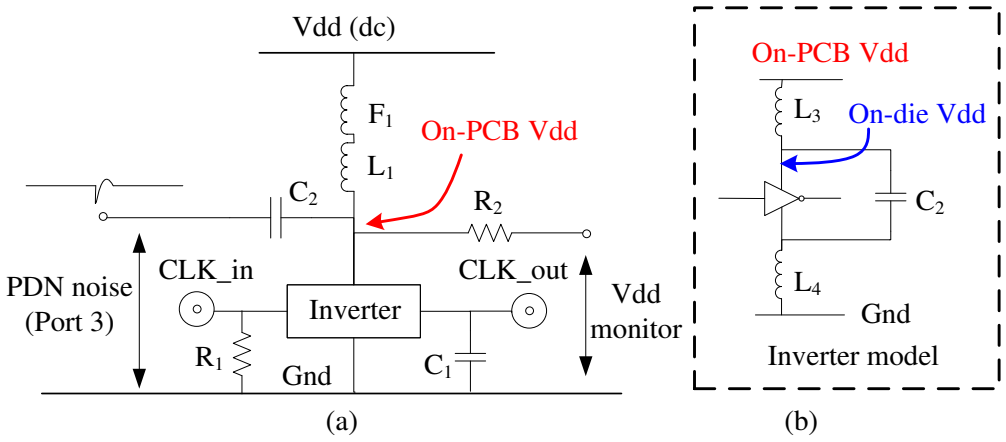

Fig. 10. (a) Circuit diagram showing connection of inverter to FPGA PDN; (b) Model of inverter package and die power delivery network

The inverter was driven with a 50 MHz clock signal while monitoring the delay of each clock edge from the input to the output of the inverter and while monitoring the power supply noise. The measured power supply voltage was also used with (23) to predict jitter. α, V_{th} and $V_{D0.ref}$ were set to 1, 0.7 and 1.5. As demonstrated in [18], although these values are estimates, the resulting model errors are expected to be low so long as the estimates are

roughly accurate, as the extracted constant A and B somewhat compensate for parameter errors.

An example of the measured and estimated delays through the inverter is shown in Fig. 11. Figure 11 shows only a portion of the signal over the PRBS7 signal. The predicted delays are within 10% of the measurement. The waveform for the predicted delays is "noisy" because of the noisy inverter power supply. Figure 12 shows the measured and predicted values of the maximum positive and negative jitter during each clock cycle. As expected, the model does a good job of predicting both the peak positive and negative jitter. The peak jitter predicted using (23) matched the measured jitter within an RMS error of 0.12 ns for the positive peak jitter and 0.19 ns for the negative peak jitter, over the entire PRBS7 sequence. The jitter repeats every 127 clock cycles with the PRBS7 sequence.

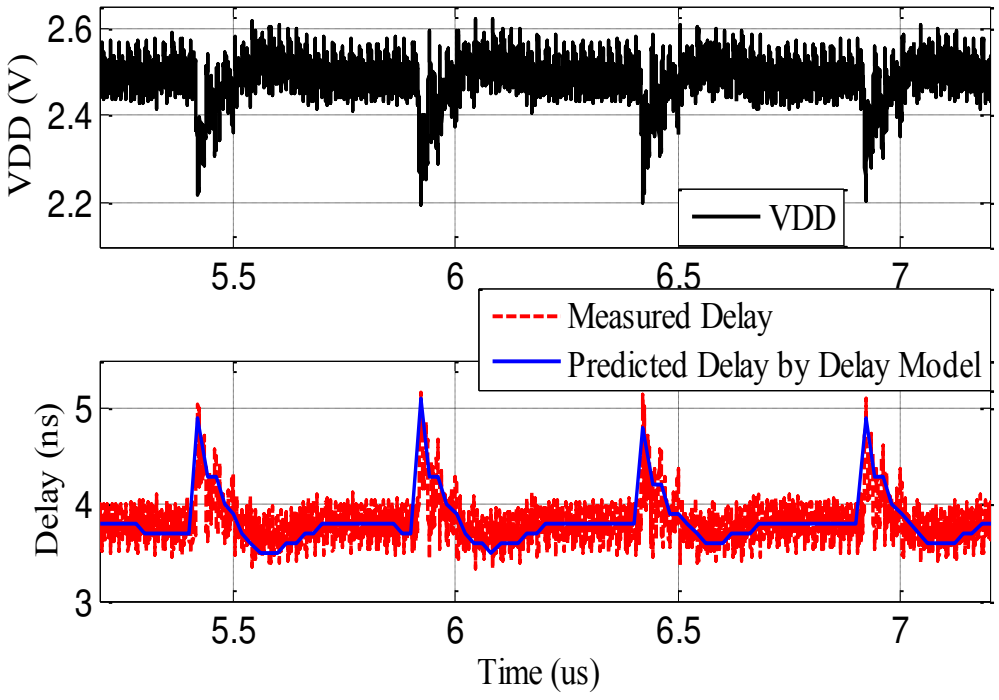

Fig. 11. Comparison of measured and estimated inverter delay

V. STATISTICAL CHARACTERISTICS OF PEAK JITTER

The mean and standard deviation of peak-to-peak jitter can be estimated directly from the statistics for the power supply noise if the relationship between jitter and noise is linear. That is, if:

$$J_{PP} \approx M \cdot (V_{DROP} - V_{PEAK}) \tag{24}$$

where M is a constant relating the power supply noise to jitter, then

$$\mu_{Jitter} \approx M \cdot \mu_{noise} \tag{25}$$

$$\sigma_{Jitter} \approx M \cdot \sigma_{noise} \tag{26}$$

where μ_{Jitter} and σ_{Jitter} are the mean and standard derivation of peak jitter due to PDN noise. The relationship between jitter and noise predicted by (16) is not linear, but it is close. Fig. 13 shows the measured values of peak negative and positive jitter as a function of the power supply noise.

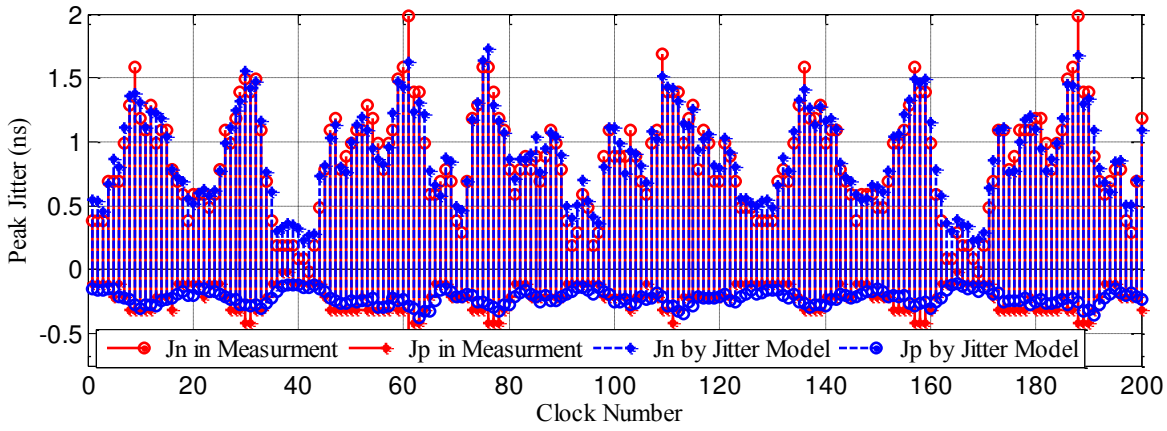

Fig. 12. Comparison of the measured peak jitter and predicted peak jitter

The measured and predicted curves for jitter are not linear, but are close, as shown by the dotted black line that was calculated using a value of M = -4.37E-9 s/V. This linear

factor can be determined directly from (16), (21), and (22) and can be used to directly estimate the mean and standard deviation of jitter from the statistics for power supply noise.

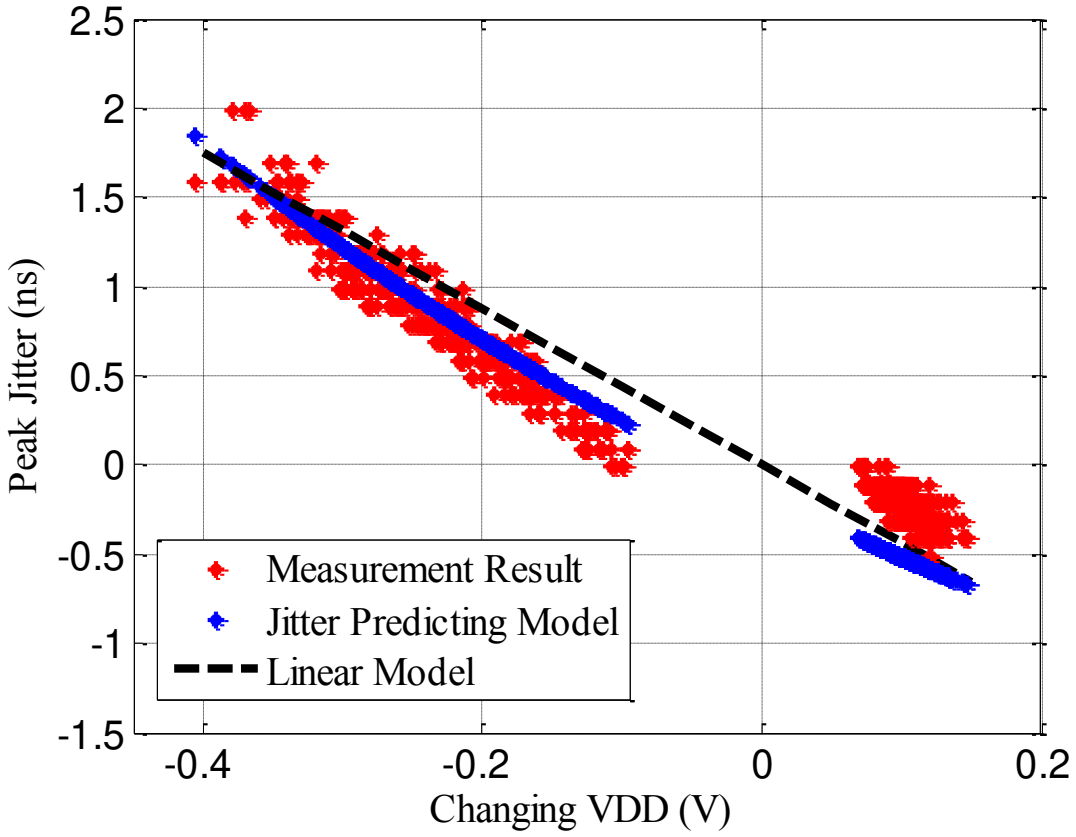

Fig. 13. Relationship between power supply noise and peak jitter

The mean and standard deviation of the peak-to-peak jitter in Table 2 were determined from measurement (i.e. from Fig. 12), were predicted using the jitter model in (16) and (23) with the measured power supply noise (also in Fig. 12), and predicted using (25) and (26) from the vectorless estimate of the statistics for power supply noise, using (12), (13), (25), and (26). The vectorless model was able to predict both the mean and standard deviation of peak-to-peak jitter within 21%.

TABLE II

STATISTICAL CHARACTERISTICS OF PEAK-TO-PEAK JITTER

Peak to Peak Jitter	Mean (ns)	Standard Deviation (ns)
Measured jitter	1.12	0.431
Jitter model with measured power supply noise	1.07	0.377
Vectorless model	1.36	0.341

VI. DISCUSSION AND CONCLUSIONS

A methodology was proposed using vectorless methods to estimate the statistical characteristics of peak power supply noise and peak jitter due to power supply noise. The mean and standard deviation of the peak power supply noise could be found within 2% and 8%, respectively, using vectorless methods. The mean and standard deviation of peak-to-peak jitter could both be found within 21%. These results are sufficient for determining the impact of a logic design on jitter. This information can be used to guide the development of the logic layout such that its current consumption is within acceptable bounds or to guide the development of defense strategies (e.g. PDN impedance) so that the current consumed by the logic will not generate unacceptable jitter. Statistically meaningful bounds can be placed on the noise or jitter using Chebyshev's inequality, which depends only on the mean and standard deviation and does not require knowledge of the underlying probability density function.

The proposed method is fast, since it relies on vectorless techniques. It is worthwhile to note that the mean and standard deviation for charge consumption need only be multiplied by a constant, determined using the methods presented here, to find the mean and standard deviation for peak-to-peak jitter. Another advantage of the vectorless method is that it allows one to relatively easily determine which portion of the circuit contributes most to the noise, and thus where to focus improvements. Since the vectorless method does not depend on knowledge of specific input data vectors, it can also be applied relatively early in the design process.

Results clearly demonstrate the accuracy of the approach when the clock speed is relatively low and the noise is on the PCB, but several extensions are needed to apply it to many other practical situations. When the clock is relatively slow, the noise contributed by one clock edge does not contribute to noise on the next clock edge. In many situations, however, the noise from one clock combines with noise from another, which can increase or decrease the overall noise. Determining the statistical characteristics of the noise in this situation is a topic for future studies.

As demonstrated in this paper, the noise waveform on the PCB is largely independent of the current waveform on the IC. This independence is an advantage when applying our approach, as the vectorless method cannot be used to determine the current waveform – only the amount of charge consumed during a clock. The waveform, however, might be needed to estimate *on-die* power supply noise, and thus the impact of switching on jitter when the logic and the clock or I/O driver share the same on-die PDN. In many modern ICs, the logic and I/O use different on-die power supplies, so the work performed here would still apply to prediction of I/O jitter in most cases.

REFERENCES

[1] J. Fan, X. Ye, J. Kim, B. Archambeault, and A. Orlandi. "Signal integrity design for high-speed digital circuits: Progress and directions." *IEEE Trans. Electromagn*. Compat 52, no. 2. pp. 392-400. 2010.

[2] P. Heydari, M. Pedram, M, "Jitter-induced power/ground noise in CMOS PLLs: a design perspective." In IEEE Computer Design, ICCD 2001. Proceedings. pp. 209-213. 2001.

[3] L. Ren, T. Li, S. Chandra, X. Chen, H. Bishnoi, S. Sun, P. Boyle, I. Zamek, J. Fan, D. Beetner, J. L. Drewniak, "Prediction of Power Supply Noise From Switching Activity in an FPGA." Electromagnetic Compatibility, *IEEE Transactions* on vol.56, no. 3, 2014.

[4] R. Burch, F. Najm, P. Yang, and T. Trick, "A Monte Carlo approach for power estimation." *IEEE Transactions on VLSI Systems*, vol. 1, no. 1, pp. 63-71, March 1993.

[5] J. Kim, L. Li, S. Wu, H. Wang, Y. Takita, H. Takeuchi, J. L. Drewniak, "Closed-form expressions for the maximum transient noise voltage caused by an IC switching current on a power distribution network". Electromagnetic Compatibility, *IEEE Transactions on*, 54(5), pp. 1112-1124. 2012.

[6] H. Shi, G. Liu, A. Liu, "Analysis of FPGA simultaneous switching noise in three domains: time, frequency, and spectrum." Proc. *DesignCon 2006*.

[7] C. T. Hsieh, J. C. Lin, S. C. Chang, "Vectorless estimation of maximum instantaneous current for sequential circuits." Computer-Aided Design of Integrated Circuits and Systems, *IEEE Transactions on*, 25(11), 2341-2352.

[8] S. Pant, D. Blaauw, V. Zolotov, S. Sundareswaran, R. Panda, R. (2003, November). Vectorless analysis of supply noise induced delay variation. In Proceedings of the *2003 IEEE/ACM international conference on Computer-aided design* (p. 184). IEEE Computer Society.

[9] L. Ren, D. Beetner, S. Sun, P. Boyle, M. Scheppers, C. Stagner, M. Deo, J. Drewniak, "Vectorless Estimation of Power Consumption Variations in an FPGA," *DesignCon 2011*.

[10]V. Laddha, M. Swaminathan, "Correlation of PDN impedance with jitter and voltage margin for high speed channels." *Electrical Performance of Electronic Packaging*. IEEE. pp. 73-76. 2008.

[11]J. Kho, C. I. Loh, B. Krsnik, Z. Li, C. S. Fong, P. Boyle, M. O. Wong, "Effect of SSN-induced PDN noise on a LVDS output buffer." In *Applied Electromagnetics*, 2007.

[12]I. Zamek, M. Wong, P. Boyle, N. Daud, L. Soh, H. Teng, C. Fong, "A study of jitter effects in nm-FPGA with respect to various physical and electrical quantities." *IEEE APACE*, Melaka, Malaysia, December 2007.

[13]H. L. Teng, S. Sun, M. O. Wong, P. Boyle, C. S. Fong, "A study of the relationship between on-chip power distribution network voltage noise, charge per clock cycle, on-chip decoupling capacitance and clock jitter in a 40-nm field programmable gate array test chip." In Applications of *Electromagnetism and Student Innovation Competition Awards* (AEM2C), 2010 International Conference on IEEE. pp. 75-79. 2010.

[14]L. N. Soh, H. L. Teng, P. Boyle, W. O. Wong, C. S. Fong, "An innovative FPGA internal core clock jitter prediction methodology." *In Electromagnetic Compatibility (APEMC)*, 2010 Asia-Pacific Symposium on IEEE. pp. 346-349. 2010.

[15]F. N. Najm, “Transition density: A new measure of activity in digital circuits.” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 310-323, Feb. 1993.

[16]T. Sakurai, and A. Richard Newton. "Alpha-power law MOSFET model and its applications to CMOS inverter delay and other formulas." *Solid-State Circuits, IEEE Journal of* 25, no. 2 (1990): 584-594.

[17]X. Gao, C. Sui, D. Beetner, S. Hemmady, J. Rivera, S. Yakura, J. Villafuerte, and D. Pommerneke. “Modeling timing variations in digital logic circuits due to electrical fast transients.” In *Electromagnetic Compatibility (EMC), 2013 IEEE International Symposium on*, pp. 484-488. IEEE, 2013.

[18]X. Gao, C. Sui, S. Hemmady, D. Pommerneke., D. Beetner, “Predicting EMI Induced Delay Errors in Integrated Circuits: Sensitivity to the Velocity Saturation Index.” *2015 Asia-Pacific International EMC Symposium and Exhibition.*

[19]H. Lan, R. Schmitt, C. Yuan, “Prediction and measurement of supply noise induced jitter in high-speed I/O interfaces.” *Proc. of DesignCon 2009.*

[20] I. Zamek, P. Boyle, Z. Li, S. Sun, X. Chen, T. Li, S. K. Chandra, “Modeling FPGA current waveform and spectrum and PDN noise estimation.” DesignCon 2008.

[21]Y. J. Zhang, G. Feng, J. Fan, “A novel impedance definition of a parallel plate pair for an intrinsic via circuit model.” Microwave Theory and Techniques, IEEE Transactions on , vol. 58, no. 12, pp. 3780-3789. 2010.

[22]C. Sui, L. Ren, D. Beetner, J. L. Drewniak, “Vectorless Prediction of Simultaneous Switching Noise from FPGAs.” *DesignCon 2015.*

III. Modeling Delay Variations of DLL Due to Power and Ground Voltage Fluctuations

Chunchun Sui, *Student Member, IEEE,* Xu Gao, *Student Member, IEEE*
David Pommerenke, *Fellow, IEEE,* Daryl G. Beetner, *Senior Member, IEEE*

Abstract—Generation and distribution of clock signals inside the IC is critical to the function of an IC. If the clock jitter is sufficiently large, it will cause timing and functional issue in the IC. Delay-locked loops (DLLs) is one of the important circuit, which widely used in multiphase clock generators, clock de-skewing circuits and clock recovery circuits. In an ideal situation, the output jitter that is created by on-chip noise can be corrected when a clean reference clock edge arrives at the DLL's input. The power and ground noise in a real circuit, however, can affect the DLL circuits' performance. This paper developed an analytical noise transfer model, which describes the power and ground noise impact on output phase disturbances in DLL. The noise transfer model can be used to predict error rate in digital circuits, can help to analyze the immunity of the DLL circuits, and could also help to develop a lower failure rate circuits. Verification of this model through simulation and measurement was also introduced in this paper.

Index Terms—Delay locked loops, noise transfer function, IC immunity

I. INTRODUCTION

High speed circuits design requires high performance from the clock synthesizers. Generation and distribution of clock signals inside the IC is one of the most important problems. The clock signals should have zero clock skew, which means all the clock signals should arrive at the inputs of registers at the same time. Otherwise clock slew will cause timing and functional issue in the IC. DLLs and Phase Locked Loops (PLLs) are usually used to generate the clock signal which is required to implement clock de-skewing circuit in RF transceiver, inter-chip communication interfaces, and clock distribution networks. In theory, PLLs are more susceptible to power supply and substrate noise because of the jitter accumulation effect, and DLLs has better immunity to on-chip noise and stability, because jitter created by the on-chip noise can be largely corrected when a clean reference clock edge arrives at the input of the DLLs. However, the power and ground noise can affect the performance of DLLs circuits, and thus impact the performance of the IC in high speed I/O and circuits in the real world.

Many books and papers investigate DLLs/PLLs. Some of them focus on the specific DLLs/PLLs design for different application. Some of them are interested in the analysis of the transfer function of different DLLs design. There are also some papers analyzing the jitter of the DLLs caused by power/ground noise. A phase-locked loop is a dynamic system that produces an output clock in response to the input clock's frequency and phase. A delay-locked loop works to achieve the same goal. This system, however, operates on a slightly different principle. It adjusts the delay of a buffer chain instead of the frequency of an oscillator. DLLs and PLLs definitions are describing in [1] - [2]. These two

books also introduced the design procedure of classical DLLs/PLLs. The phase detector, charge pump and loop filter are each used in both DLLs and PLLs. The difference between them is that DLLs contain a Voltage Control Delay Line (VCDL), and PLLs contain a Voltage Control Oscillator (VCO). Author in [3] starts with the simplest DLLs model, and then describes design considerations and techniques to achieve high performance, such as avoiding false lock, maintaining 50% clock duty cycle and building unlimited phase range for frequency synthesis.

Transfer functions are typically used to describe the behavior of DLL and PLL during normal operations, in terms of the relationship between output phase and input phase in the frequency domain. Thus, the transfer function reveals the influence of the input jitter on the output jitter. The transfer function of DLL circuits is commonly considered as first-order system. From the basic theory of DLLs, the closed loop transfer function is first-order transfer function, however, in the real circuit, parasitic capacitance, decoupling capacitance and other effects can cause the DLLs system to be a second-order system or even higher order. A second-order DLLs in z domain was developed in [4]. This paper shown that in a widely used DLL configuration, jitter peaking always exists. Additionally, high-frequency jitter does not get attenuated, as previous analyses suggested. Author in [5] also introduced a low power 3.125 Gb clock and data recovery circuit with a second-order clock and data recovery circuit. The circuits was performed the interpolation which is capable of tracking frequency offsets while exhibiting low phase wander. Author in [6] analyzes the affects of the transmitter and receiver phased-locked loop (PLL) phase noise, which translates to time-domain clock/data jitter, on the performance of high-speed transceivers. Analytical expressions are derived to incorporate both transmitter and

receiver clock jitter into serial link operations. These papers achieved a good balance between the design requirements and design costs. All these papers helped increase the knowledge base regard to DLL circuits.

Jitter due to power/ground supply voltage fluctuation has been studied recently. This work is usually represented by DLLs noise transfer function, which describes the relationship between output phase and power/ground supply voltage. It is challenge to develop a second-order noise transfer functions. Some papers described the noise transfer function as a first order system with assumption that all the components are working in linear mode. The assumption is not always true when it comes to the real world. Few papers talk about the relation between the noise transfer function parameters and the DLLs circuits. Substrate noise effects on the performance of DLLs were studied in [4] . This paper proposed a stochastic model that it can be used for the substrate noise. This model was then utilized to derive the phase noise of the VCDL inside the loop. The model was based on an ideal differential delay stage. Thus this model cannot be easily applied to real design circuits. Author in [7] and [8] introduced DLL jitter affected by power supply noise on power distribution network, and also pointed out that the jitter is related to transfer impedance of a hierarchical PDN using Transmission Line Matrix method. The PDN transfer impedance was simulated from the port where the circuits is located in the PCB to the power input port. Large noise was transferred with low transfer impedance. This noise produced large jitter. This paper came to the conclusion that low inductance of hierarchical PDN will decrease the DLL output peak to peak jitter. This paper did not specifically give a transfer function between power noise at the input and jitter at the output. A model of the delay variations in CMOS digital logic circuits due to electrical disturbances in the power

supply in the time domain was developed in [9] . This model worked well for circuits like combinational logic and a ring oscillator. This model needs to be developed further before it can be applied to analog circuits with negative feedback, like DLLs/PLLs. Unlike the previous research on DLL output peak to peak jitter in frequency domain, author in [10] describes both the on-chip measurement of jitter transfer as well as the supply sensitivity transfer function of PLLs and DLLs. The procedure for estimating the frequency-domain transfer functions from the measured time-domain responses was outlined. The measured jitter transfer function describes the relation between output jitter and input jitter. Supply sensitivity function describes the relation between output jitter and power supply voltage. These two transfer functions of the PLL/DLL provide helpful insights on the major sources of the clock jitter. However, no analytical equation is developed to describe the jitter change due to input clock phase or its supply voltage in this paper. In addition, the analysis only works for a design which has already been implemented. An analytical model with its physical meaning will help a lot in the pre-design level. In [7] and [11], a new model is proposed to estimate the affect of simultaneous switching noises (SSNs) on a DLL's clock jitter in a hierarchical system of chip, package and PCB. This method was used to investigate the SSN coupling paths and their impact on clock jitter. These papers developed SSNs to output jitter transfer function in frequency domain. Their measurement results, however, did not closely match the transfer function.

In previously published work, DLL noise transfer functions describe the output phase change tendency alone with power ground noise in frequency domain. These transfer functions did not well match the simulated results and the experimental results. It is even challenging to develop an accurate DLL delay/jitter model in time domain with

power/ground fluctuation condition. A novel and accurate delay model for DLL is proposed in the following thesis, which can describe the output phase disturbance in the present of the large power supply variations. The proposed analytical delay model is used to predict the output delay variation when the power/ground plan is injected a sinusoidal waveform or EFTs. To describe the DLL delay model due to power ground voltage fluctuation, the DLL circuits design is first introduced in the following section.

The paper is presented in five sections. Section I is introduction, which presents the background of the research on DLLs/PLLs. The delay locked loops circuits design is studied in Section II. Section III introduces the derivation of the DLLs delay model, where each component delay modeling is targeted for and verified. In Section IV, the model is applied to the real IC measurement. Discussion and conclusions are given in Section V.

II. DELAY LOCKED LOOP DESIGN

Based on the application, DLLs circuits can be classified into two catalogs, named Type I and Type II ([12]). In a Type I DLLs, the reference is compared with the delayed version of itself. This architecture is widely used in DLL-based frequency synthesizers, multiphase clock generators and clock de-skewing circuits.

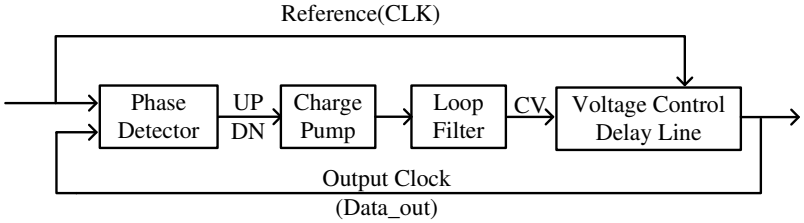

Fig. 1. Type I DLLs structure

As the figure shows, Fig. 1 and Fig. 2 illustrate a type 1 DLL's circuit structure and its time domain simulation results. A DLL's basic function is to decrease the phase difference between CLK and Data_out to either zero or some other predefined value. The process is begun when phase detector detects the initial phase difference between Data_out and CLK. This phase difference is revealed at the UP/DN signal at the initial time. The active UP/DN signal pulls up/down the control voltage (CV), and the CV signal decreases/increases the phase difference between CLK and Data_out. At this point, both the CLK and the Data_out are finally locked.

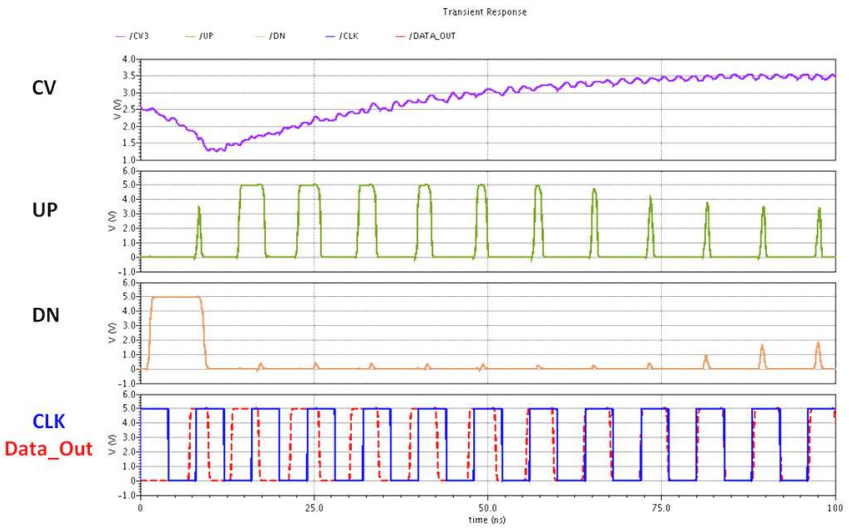

Fig. 2. Function simulation results of Type I DLLs

In a Type II DLL, illustrated in Fig. 3 and Fig. 4, the reference is compared to the delayed version of an uncorrelated signal. Typically, Vin and CLK carry the same frequency signal and different phases. The initial phase difference between CLK and Data_out can be adjusted by changing Vin's delay. This initial phase difference cannot, however, change in the Type I DLL structure. This is the most important characteristic that differentiates one DLL structures from another. At the beginning Data_out signal in the

simulation results led the CLK signal. Thus, the UP signal was active, and the pulse width of the UP signal was equal to the phase difference between CLK and Data_out. The negative feedback system decreased this phase difference at each clock cycle, as illustrated by the UP signal's decreasing pulse width. This phase difference was decreased to zero after several clock cycles. Both the Data_out and the CLK signal were in phase, and the system was in its stable state. The type II DLL's architecture was widely used in DLL-based clock recovery circuits.

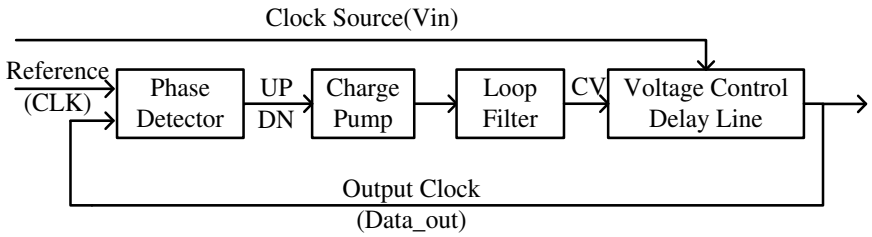

Fig. 3. Type II DLLs structure

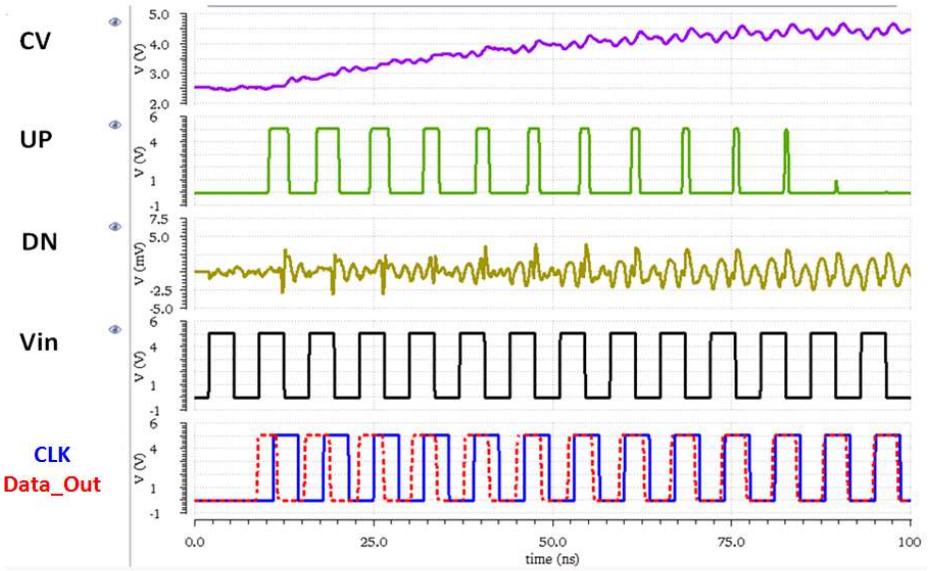

Fig. 4. Function simulation results of Type II DLLs

The following sections will introduce the phase detector, charge pump, loop filter

and voltage control delay line, as illustrated in Figs. 1 and Fig. 3. These four circuits are the basic components of a DLL.

A. Phase Detector

The first stage of a DLL is the phase detector. Two common types of phase detectors, an XOR gate and a phase frequency detector (PFD), have significantly different performance capabilities and limitations.

The XOR phase detector is, essentially, an XOR gate that can implement the a phase detector's function. The pulse width of logic high at the output of an XOR gate indicates the phase difference between the two input signals. One benefit of an XOR phase detector is that the circuits are easy to implement. The other benefit is that an XOR gate takes smaller design area compared with a PFD. Unfortunately, the limitation of an XOR phase detector is that the output of an XOR gate cannot specify if the clock leads or lags the data.

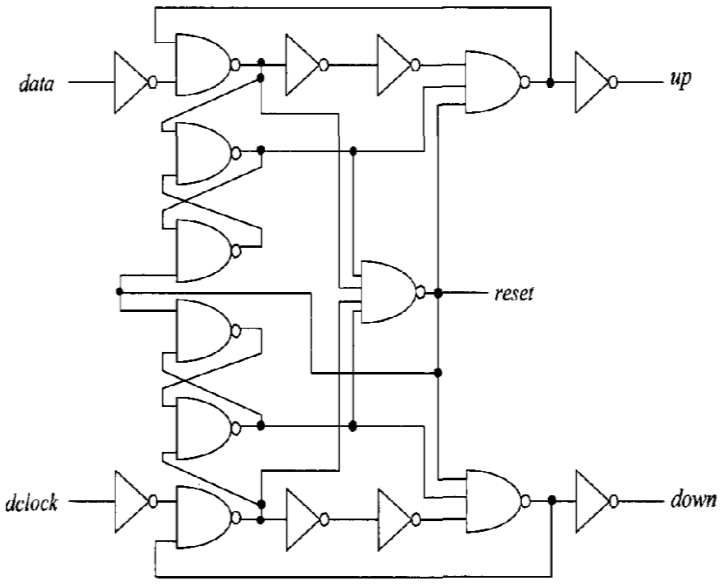

Fig. 5. Circuit schematic of PFD

A phase frequency detector is comprised of two D flip-flops, as shown in Fig. 5. The output of a PFD depends on both the phase and frequency of the input, as shown in a simulated example in Fig. 6. Unlike the XOR phase detector, a PFD does not depend on the pulse width of input signals A and B. If the A rising edge lags the B rising edge, the "DN" output of the phase detector goes high, while the "UP" output of the phase detector remains low. When the A signal leads the B signal, DN remains low, while UP goes high a time equal to the phase difference between A and B. Note that when the DLLs is in its locked state, both UP and DN remain logically low.

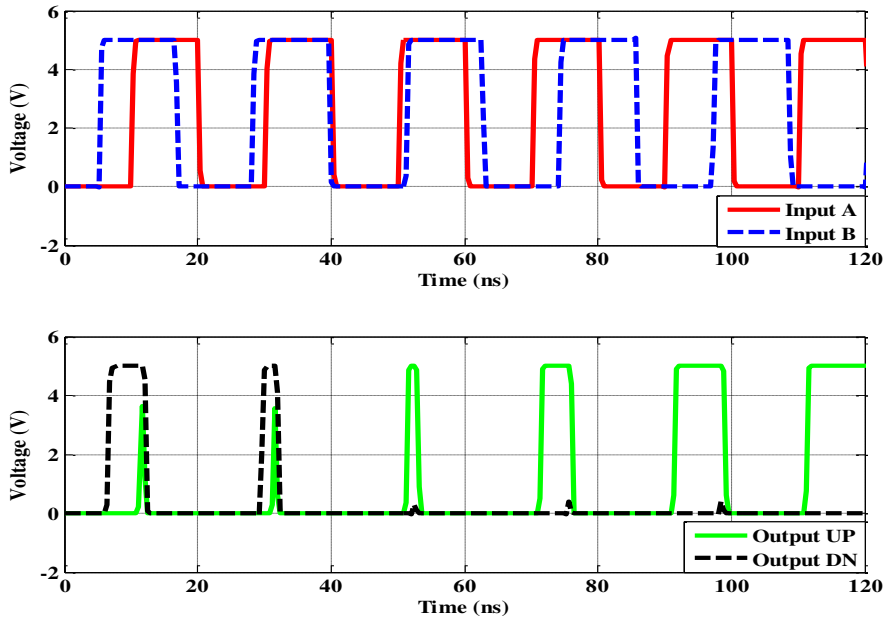

Fig. 6. Functional simulation of PFD

The circuits within a PFD are more complex than those in an XOR phase detector, The circuits within a PFD take more design area in CMOS technology. Nevertheless, the PFD is widely used in DLL and PLL designs, because it has better performance than XOR phase detector. Based on the analysis above, the DLLs in this study were designed with a PFD.

B. Charge Pump

The charge pump circuit, as depicted in the schematic shown in Fig. 7, was used to combine both the UP signal and the DN signals into an analog control voltage. When PFD UP signal goes high, the M5 turns on, connecting the current source to the next stage loop filter. The output of the charge pump circuits, also known as the control voltage goes up. Similarly, when the DN signal goes high, M8 turns on, the charged loop filter goes through M8 and M7 to ground, so the control voltage drops.

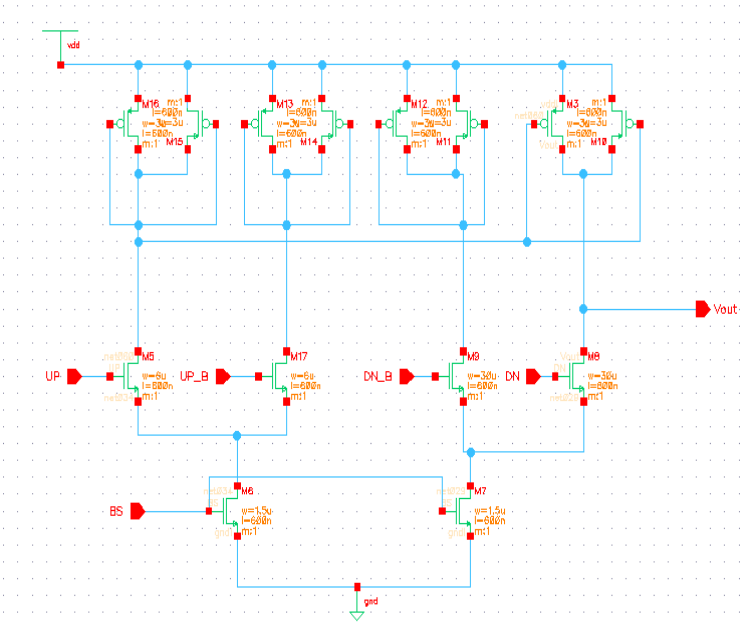

Fig. 7. Circuit schematic of charge pump

C. Loop Filter

A loop filter is an important component in DLLs design, because it determines the speed of DLL adjustment based on the phase error.

One way of making capacitors is to use the two poly silicon layers or two metal layers. The benefit of this method is the capacitor value can be well controlled, and the shortcoming is the capacitor takes large design area. It is even impossible to implement

large capacitor in this method. The other way would be to use the gate oxide and actually build a transistor whose gate area (W x L) would actually give us the capacitance. These are called MOS capacitors, and they only work properly when the transistor is strongly inverted or depleted. Otherwise, the capacitance can vary with the voltage across it. Since the our designed capacitor value is not large, two metal layers method is used in this DLL design, the loop filter is a capacitor connecting with control voltage and the Vdd, and the capacitor value is 300fF.

D. Voltage Control Delay Line

Voltage Control Delay Line (VCDL) is an important component in DLL system. Figure 8 is an illustration of two basic implementations of the VCDL delay cell. Since delay is a determined by load capacitance and drive resistance, it can be varied by adjusting either one. Figure 8 (a) is called a current starved inverter delay cell. It is possible to regulate the delay of this element by control the overcharging current of the output parasitic capacitor, and the control voltage (CV) can control the value of the current source created by the M0 and M4 transistors.

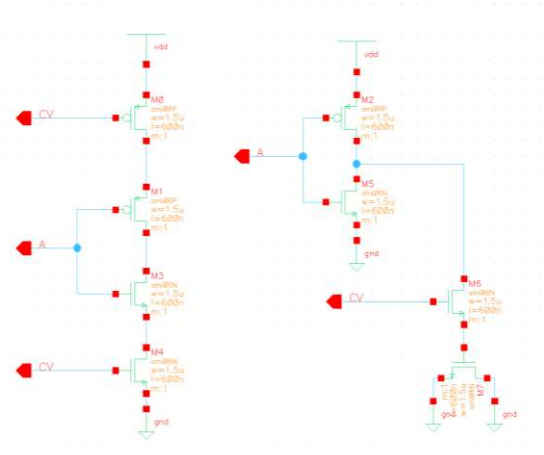

Fig. 8. (a) Resistance delay cell in VCDL. (b) Capacitance delay cell in VCDL

Figure 8 (b) shows a structure known as shunt capacitor delay cell. A VCDL creates a chain of capacitive loaded inverters, and the load capacitors (transistors M7) are connected to the outputs of the inverters only by NMOS transistor M6. The transistor M6 serves as a switch and transistor M7 serves as capacitor. The changing control voltage effectively results in the changing of inverter load capacitance. The load capacitance is related to the propagation delay of the inverter. That is the basic mechanism of VCDL circuits. This paper adopts the shunt capacitor delay cell in the VCDL, because the control voltage can control a large delay range compared to current starved inverter delay cell.

III. DELAY MODEL FOR DLLS CIRCUITS

In this section, a delay model of DLLs circuits is developed based on a well know inverter delay model. The power/ground plan voltage fluctuation is also included in the delay model. This model can well describe the output phase change caused by the power/ground noise, which helps to explain the jitter variation in DLLs circuits.

A. Inverter Delay Model

Because the basic schematic of a VCDL is an inverter chain with transistor load structure, the inverter propagation delay is firstly reviewed.

The propagation delay of an inverter is defined as the time between input signal reaching half of V_{dd} to the output signal reaching half of V_{dd} in the inverter circuits. The propagation delay t_p for a single inverter was developed in [13] and [9], as Fig. 9 shown,

$$t_p = \left(\frac{1}{2} - \frac{1-\frac{V_{th}}{V_{dd}}}{1+\alpha}\right)\cdot t_T + \frac{C_L V_{dd}}{2I_{D0}} \tag{1}$$

$$I_{D0} = (\frac{V_{dd} - V_{th}}{V_{dd,ref} - V_{th}})^{\alpha} I_{D0,ref} \quad (2)$$

where V_{th} is the threshold, V_{dd} is the power supply voltage , α is the velocity saturation index for a MOSFET (typically from 1 to 2), t_T is the rise or fall time of the input signal, I_{D0} is the drain current when $V_{GS} = V_{DS} = V_{dd}$, and C_L is the output capacitance driven by the gate. $I_{D0,ref}$ is the drain current when $V_{GS} = V_{DS} = V_{dd,ref}$.

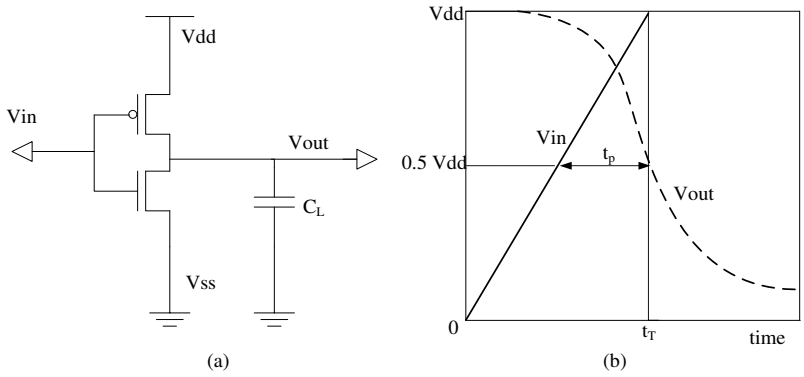

Fig. 9. (a) Inverter schematic. (b) Inverter time domain waveform

The high-to-low and the low-to-high propagation delay times are dependent on the parameters for the nFETs and pFETs, respectively. In the real circuit design, in order to balance the inverter's pull up capability and pull down capability, the nFETs and pFETs are designed with similar parameters, which means that high-to-low and low-to-high propagation delay times can typically be assumed identical in the inverter chain circuits.

In many cases, it is difficult to know the rise time of the input signal at each stage, so an assumption is made that the propagation delay at each stage is approximately equal to half of the rise time. This assumption is based on the inverter chain having same inverter cell and for each stage, the load capacitance are also same. With this assumption, equations (1) and (2) can be rewritten as

$$t_p = \frac{C_L V_{dd}}{2I_{D0}} \cdot \frac{1+\alpha}{2(1-\nu_T)} = \frac{C_L V_{dd}}{4(\frac{V_{dd}-V_{th}}{V_{dd,ref}-V_{th}})^{\alpha} I_{D0,ref}} \cdot \frac{1+\alpha}{\frac{V_{dd}-V_{th}}{V_{dd}}} \quad (3)$$

Simplified version for equation (3) is:

$$t_p = \frac{C_L}{4}(1+\alpha)\frac{(V_{dd,ref}-V_{th})^{\alpha}}{I_{D0,ref}} \frac{V_{dd}^2}{(V_{dd}-V_{th})^{\alpha+1}} \quad (4)$$

This model can predict the propagation delay of inverter chain circuits, [9] shows very good results.

B. Voltage Control Delay Line Delay Model

The basic schematic of a VCDL, as shown in Fig. 10, is an inverter chain with transistor load structure. The control voltage controls transistor M3 operation, and the function of transistor M4 is a capacitor. However, the model (4) cannot directly apply for the VCDL delay cell, because the load capacitance of this 'inverter chain' is not a constant value, the load capacitance is related to the control voltage. The model should take into account the shunt capacitor MOSFET M4 and voltage control MOSFET M3 in Fig. 10.

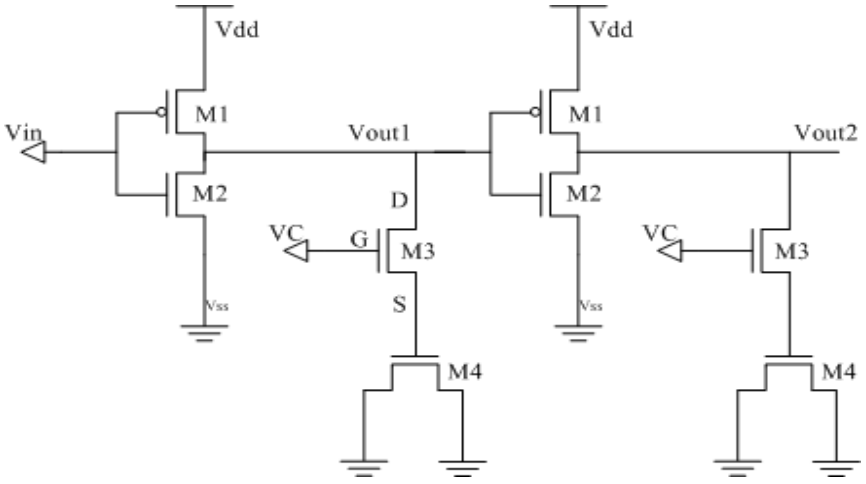

Fig. 10. VCDL delay cell schematic (Two stage)

VCDL delay cell equivalent circuits were developed as shown in Fig. 11. The Control Voltage controls MOSFET M3, and M3 works as a switch. C_{Load1} and C_{Load2} are the load capacitance due to the diffusion in the output FETs, routing and downstream gate oxide. C_{Control1} and C_{Control2} are the capacitance associated with M3 source to substrate capacitance and the M4 gate to body capacitance. When Vgs-Vth<Vds, the MOSFET M3 is turned off, the state of switch is open. The load capacitance is C_{Load1} and C_{Load2} at each output stage. When Vgs-Vth>Vds, the MOSFET M3 is turned on, and the state of the switch M3 is closed. The overall load capacitance is $C_{\text{Load1}} + C_{\text{Control1}}$ and $C_{\text{Load2}} + C_{\text{Control2}}$ at each output stage.

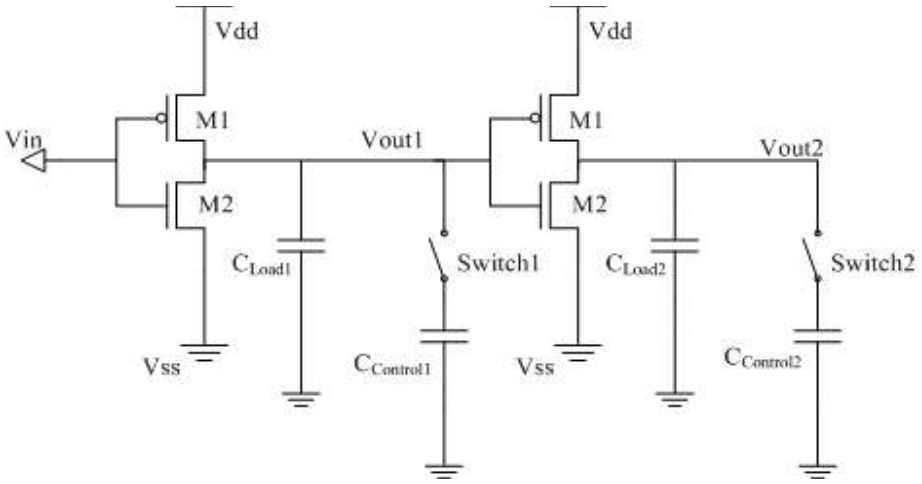

Fig. 11. VCDL delay cell equivalent circuits schematic (Two stage).

To determine a model for the propagation delay, the output Low-to-High transition is analyzed first, as shown in the Fig. 12. For each control voltage, the output signal first charges with $C_{\text{Load1}} + C_{\text{Control1}}$ and then charges with C_{Load1}. The control voltage controls the time instant when the switch occurs. For example, in the schematic shown in in Fig. 11, firstly, the output is charging with overall load capacitance. In this period, M3 transistor is turned on. The overall load capacitance is the load capacitance in M2 and M4

transistor capacitance. When the output is charged to switching point, M3 transistor is turned off, and the output is charged with load capacitance in M2 to Vdd. The voltage difference between control voltage and switching point is a threshold voltage. The control voltage is higher, the switching point is higher. This causes the propagation delay to vary with the control voltage. In order to analyze the propagation delay time, which is defined as time it takes for the output to reach half of Vdd, switching point is expressed as V_{SW}, and the switching point at half of Vdd is defined as V_{SW0}.

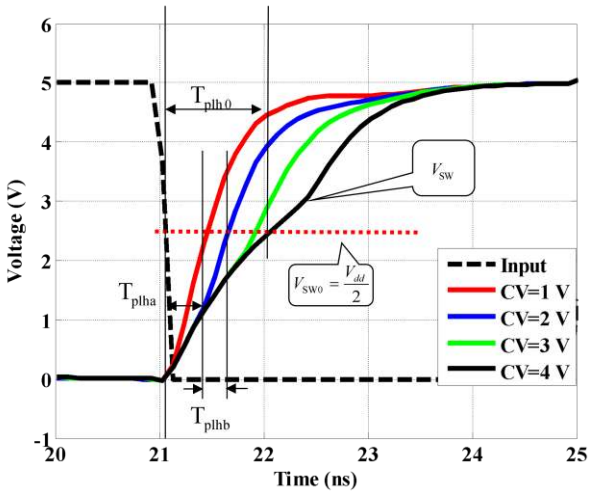

Fig. 12. VCDL delay cell Low-to-High transition simulation results (First stage).

When $V_{Control} - V_{th} > V_{SW0}$, switching point V_{SW} is above V_{SW0}. The propagation delay, Tplh0, as shown in Fig. 14, is constant, because the control voltage changes the output waveform above half of Vdd. The constant delay Tplh0 is related to charging time on capacitance $C_{Load1} + C_{Control1}$.

When $V_{Control} - V_{th} < V_{SW0}$, the switching point V_{SW} is below V_{SW0}. The delay changes with the control voltage. The propagation delay comes up with two parts: Tplha and Tplhb,

where Tplha is related to the charging time on capacitance $C_{\text{Load1}} + C_{\text{Control1}}$, and Tplhb, related to the charging time on capacitance C_{Load1} .

The How-to-Ligh transition is also analyzed with similar mechanis, but follows the opposite procedure compare to Low-to-High transition.

Because the VCDL is an inverter chain connection, analyzing the propagation delay in units is reasonable. Each unit contains two inverters. That is to say, the propagation delay in a unit includes a low-to-high transition and also a high-to-low transition. According the above analysis, the control voltage will always impact the capacitance seen in either the low-to-high or the high-to-low switching event, but not both. The overall propagation delay in a unit can be assumed constant with a fixed control voltage, so an equivalent circuit that consists of a two stage inverter chain with a load capacitor can be found.

The equivalent load capacitor can be found using the law of conservation of charge to determine the relation between the control voltage and equivalent capacitor. In the output Low-to-High transition process, the load capacitor is charging. The total charge present on the capacitor is given by

$$Q_C = V_{sw}(C_{\text{load1}} + C_{\text{Control1}}) + (V_{dd} - V_{sw})C_{\text{load1}} \tag{5}$$

Simplifying equation (5), gives

$$Q_C = V_{sw}C_{\text{Control1}} + C_{\text{load1}}V_{dd} \tag{6}$$

Similarly, in the output High-to-Low transition process, the load capacitor is discharging. The total charge on the capacitor is given by

$$Q_D = (V_{dd} - V_{sw})C_{\text{load1}} + V_{sw}(C_{\text{load1}} + C_{\text{Control1}}) \tag{7}$$

Simplify equation (7), gives

$$Q_D = V_{sw} C_{\text{Control1}} + C_{\text{load1}} V_{dd} \tag{8}$$

Switching voltage V_{SW} is also the MOSEFT M3 drain voltage and control voltage $V_{Control}$ is the MOSFET M3 gate voltage. The switching voltage follows control voltage with a difference of a threshold voltage, so the relation between the two voltages are described by

$$V_{sw} = V_{control} - V_{th} \tag{9}$$

Equivalent capacitor is defined as C_{EQ}. The law of conservation of charge requires

$$Q_C = Q_D = V_{dd} C_{EQ} \tag{10}$$

Substituting equation (6) and equation (8) into equation (10), the relation between C_{EQ} and V_{SW} is

$$C_{EQ} = \frac{V_{sw}}{V_{dd}} C_{\text{Control1}} + C_{\text{load1}} \tag{11}$$

Substituted equation (9) into equation (11), the load capacitance is given by

$$C_L = \frac{V_{Contol}(t) - V_{th}}{V_{dd}(t)} C_{\text{Control1}} + C_{\text{load1}} \tag{12}$$

Equation (12) shows the final linear relation between the equivalent capacitor and control voltage with a constant Vdd, and this equation also shows that the linear relation changes if Vdd changes.

Substituted equation (12) into equation (3) and (4), with given equation in [13] and [9], analytical delay model of VCDL is shown by

$$\begin{aligned} t_p(t) = {} & \frac{1}{I_{D0,ref}} \left(\frac{V_{Contol}(t) - V_{th}}{V_{dd}(t)} C_{\text{Control1}} + C_{\text{load1}} \right) \\ & \left(\left(\frac{1}{2} - \frac{V_{dd}(t) - V_{th}}{1+\alpha} \right) \cdot \left(\frac{0.9}{0.8} + \frac{QV_{D0,ref}}{0.8V_{dd}(t)} \ln \frac{10QV_{D0,ref}}{eV_{dd}(t)} \right) + \frac{V_{dd}(t)}{2} \right) Q^2 \end{aligned} \tag{13}$$

where $Q = \frac{(V_{dd}(t) - V_{th})^{\alpha/2}}{(V_{dd,ref} - V_{th})^{\alpha/2}}$. This equation gives the propagation delay through the VCDL in terms of both the control voltage and the power supply voltage.

To verify the VCDL analytical delay model in equation (13), the parameters need to be calculated first. C_{Load1} and C_{Control1} can be extracted from the simulation. Fig. 10 and Fig. 11 show the simulation schematic. First of all, a low frequency clock signal is used as input signal at both circuits. A constant control voltage is used in VCDL delay cell in Fig. 10. The VCDL output low-to-high transition is recorded as shown in Fig. 20 blue curve. Then, simulation schematic in Fig. 11 with $C_{\text{Load1}} + C_{\text{Control1}}$ is performed. By changing the capacitor $C_{\text{Load1}} + C_{\text{Control1}}$ value, green curve is founded to fit the first part of the VCDL output. The schematic in Fig. 11 with C_{Load1} only is simulated. By changing the capacitor C_{Load1} value, red curve are founded to fit the second part of the VCDL output. In this process, C_{Load1} and C_{Control1} is predicted with a constant control voltage. C_{Load1} and C_{Control1} are independent of the control voltage and power ground voltage. In another word, C_{Load1} and C_{Control1} are constant value for a specific design, it can be used to predict the propagation delay with variable control voltage and variable power ground voltage.

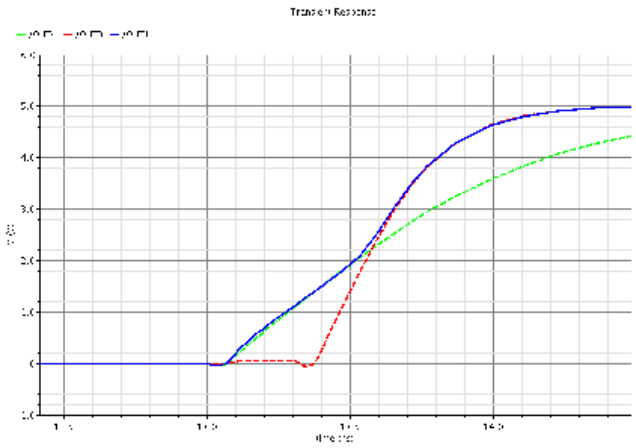

Fig. 13. VCDL load capacitor extraction simulation results.

VCDL propagation delay model (13) was verified with simulations. Fig. 10 shows the simulation schematic. With the simulation, Vdd and Control voltage are injected with different noise sources. The simulation result in Fig. 14 shows a positive EFT noise is injected into Vdd pin and a sine wave is injected into control voltage. Input wave and output wave are shown on the top figure. The logic high of output waveform follows the same trend of the Vdd signal. Positive EFTs noise will cause the logic high of output signal to go above the normal power supply voltage. Matlab code is programmed to calculate the propagation delay change by these noise injections. At the same time the propagation delay is also calculated by the analytical model using (13). That is to say, the simulation result in the delay change is based on the output waveform only, and the analytical model delay is calculated based on the Vdd and control voltage waveforms.

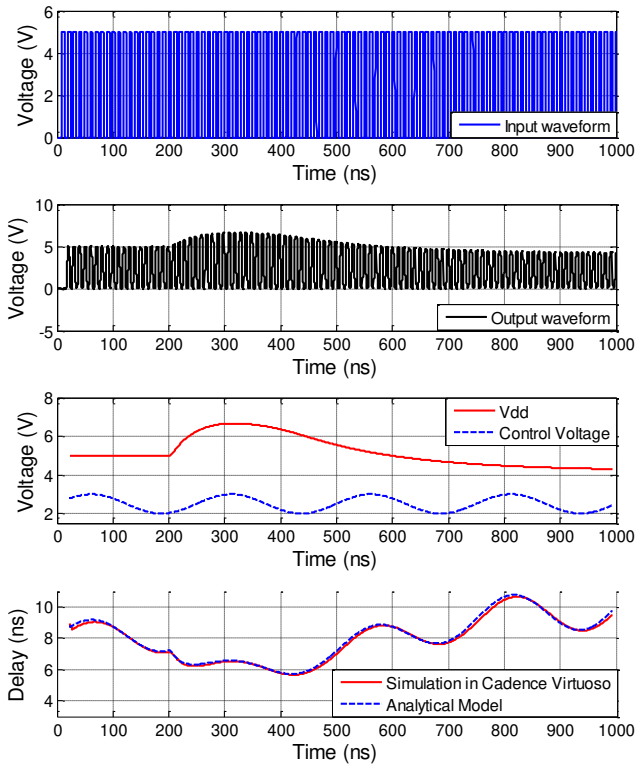

Fig. 14. VCDL delay model verification results. A positive EFT noise is injected into Vdd pin and a sine wave is injected into control voltage

The simulation results match with the VCDL propagation delay model with a RMS error of 0.12 ns, which is 2.67% error rate. This result demonstrates the accuracy of the proposed VCDL propagation delay model.

C. Voltage Control Delay Line Long-term Jitter Model

Jitter caused by the power/ground voltage fluctuation is investigated. Long-term jitter is defined as the time variations of a digital signal's significant instances from their ideal positions over many clock cycles. Based on the VCDL propagation delay model, the Long-term jitter value of the VCDL circuit can be predicted by

$$Jitter_{PP\_VCDL} = \max(t_p(t)) - \min(t_p(t)) \tag{14}$$

This Long-term jitter value determined the performance of the VCDL circuit. If the voltage fluctuation induced peak to peak jitter exceed half of the clock cycles, the eye diagram will be closed and the error will occur in the system.

D. DLL Close Loop Noise Transfer Function

The transfer function noise injection and VCDL can be given by:

$$\Phi_{out}(\omega) = G(\omega) + G_N(\omega) \tag{15}$$

$$G(\omega) = \Phi_{in}(\omega) + G_{DET}(\omega) \cdot G_{FIL}(\omega) \cdot G_{VCDL}(\omega) \cdot \frac{2\pi}{T_{clk}} \cdot \Phi_{out}(\omega) \tag{16}$$

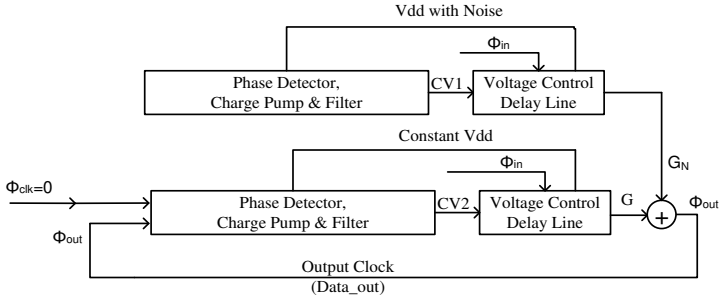

Fig. 15. DLL schematic with noise injection

Set $\omega_{clk} = \frac{2\pi}{T_{clk}}, \Phi_{in}(\omega) = 0$, assume the input clock signal is ideal clock source without phase variation. So

$$\begin{aligned} &\Phi_{out}(\omega) - G_{DET}(\omega) \cdot G_{FIL}(\omega) \cdot G_{VCDL}(\omega) \cdot \frac{2\pi}{T_{clk}} \cdot \Phi_{out}(\omega) \\ &= G_N(\omega) \end{aligned} \tag{17}$$

This equation can be simplified as

$$\frac{\Phi_{out}(\omega)}{G_N(\omega)} = \frac{1}{1 - G_{DET}(\omega) \cdot G_{FIL}(\omega) \cdot G_{VCDL}(\omega) \cdot \omega_{clk}} \tag{18}$$

where

$$\begin{aligned} &G_{DET}(\omega) = K_{DET} = -\frac{I_{pump}}{\pi} = -\frac{100uA}{\pi} \\ &G_{FIL}(\omega) = \frac{1}{j\omega C} \\ &G_{VCDL}(\omega) = K_{VCDL} = 1400\,ps/V \end{aligned} \tag{19}$$

set $a = K_{VCDL} \cdot \frac{2 \cdot I_{pump}}{T_{clk} \cdot C}$

So

$$\frac{\Phi_{out}(\omega)}{G_N(\omega)} = \frac{s}{s+a}, \tag{20}$$

Then

$$\Phi_{out}(\omega) = \frac{s}{s+a} \cdot G_N(\omega) \tag{22}$$

Convert this equation into time domain, the final DLL noise transfer function is given by

$$\Phi_{out}(t) = \left(\delta(t) - a \cdot e^{-at}\right) * g_n(t) \tag{23}$$

where $g_n(t) = f(V_{dd}(t))$

IV. CONCLUSION AND FUTURE WORK

An analytical delay model of VCDL was proposed to predict propagation delay variations when the power supply is disturbed by an electromagnetic event. Simulated results demonstrate the accuracy of the VCDL delay model. The VCDL analytical delay model is the key part of the overall delay model in DLLs. Similarly, analytical delay models for other DLL components were developed. Then, overall DLL analytical delay model was developed. The proposed analytical delay model is used to predict the output delay variation when the power/ground plan is injected a sinusoidal waveform or EFTs.

The future work includes the jitter prediction model and the verification of the overall delay/jitter model in simulation and also measurement.

REFERENCES

[1] N. H. E. Weste and D. Harris, CMOS VLSI Design: A Circuits and Systems Perspective, 4th ed. Boston: Pearson, 2010.

[2] Baker, R. Jacob. CMOS: circuit design, layout, and simulation. Vol. 18. Wiley-IEEE Press, 2011.

[3] Yang, Chih-Kong Ken. "Delay-locked loops-an overview." Phase-Locking in High-Peformance Systems, Wiley-IEEE Press, New York, NY (2003): 13-22.

[4] Heydari, P. (2002, August). Analysis of DLL jitter due to substrate noise. In Circuits and Systems, 2002. MWSCAS-2002. The 2002 45th Midwest Symposium on (Vol. 1, pp. I-348). IEEE.

[5] Ng, H. T., Farjad-Rad, R., Lee, M. J., Dally, W. J., Greer, T., Poulton, J., Senthinathan, R. (2003). A second-order semidigital clock recovery circuit based on injection locking. Solid-State Circuits, IEEE Journal of, 38(12), 2101-2110.

[6] Hanumolu, P. K., Casper, B., Mooney, R., Wei, G. Y., & Moon, U. K. (2003). Analysis of PLL clock jitter in high-speed serial links. Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on, 50(11), 879-886.

[7] Shin, M., Shim, J., Shim, Y., & Kim, J. (2010, April). Investigation of experimental verification for various Power Distribution Network cases through DLL Clock jitter affected by SSN. In Electromagnetic Compatibility (APEMC), 2010 Asia-Pacific Symposium on (pp. 325-329). IEEE.

[8] Shin, M., Yoon, C., Cho, J., Shim, J., Shim, Y., & Kim, J. (2009, July). Analysis of DLL Jitter Affected by Power Supply Noise on Power Distribution Network. In 2009 International Symposium on Electromagnetic Compatibility, Kyoto (pp. 20-24).

[9] Gao, X., Sui, C., Beetner, D., Hemmady, S., Rivera, J., Yakura, S., Pommerneke, D. (2013, August). Modeling timing variations in digital logic circuits due to electrical fast transients. In Electromagnetic Compatibility (EMC), 2013 IEEE International Symposium on (pp. 484-488). IEEE.

[10] Kim, J. (2009). On-chip measurement of jitter transfer and supply sensitivity of PLL/DLLs. Circuits and Systems II: Express Briefs, IEEE Transactions on, 56(6), 449-453.

[11] Shim, Y., Bae, B., Koo, K., & Kim, J. (2011, November). Modeling of simultaneous switching noise effects on jitter characteristics of delay locked loop in a hierarchical system of chip-Package-PCB. In Electromagnetic Compatibility of Integrated Circuits (EMC Compo), 2011 8th Workshop on (pp. 188-193). IEEE.

[12] Lee, M-JE, William J. Dally, Trey Greer, Hiok-Tiaq Ng, Ramin Farjad-Rad, John Poulton, and Ramesh Senthinathan. "Jitter transfer characteristics of delay-locked loops-theories and design techniques." Solid-State Circuits, IEEE Journal of 38, no. 4 (2003): 614-621.

[13] Sakurai, Takayasu, and A. Richard Newton. "Alpha-power law MOSFET model and its applications to CMOS inverter delay and other formulas." Solid-State Circuits, IEEE Journal of 25, no. 2 (1990): 584-594.

SECTION

2. CONCLUSION

In the first paper, a method was proposed to estimate the pdf of random jitter from measurements of total jitter. This result can then be used to estimate the contribution of deterministic jitter (which includes crosstalk induced jitter). A method was also proposed for estimating crosstalk induced jitter from measurements of total jitter. The use of a Wiener filter allows the estimate to be made accurately even in the presence of measurement noise. The crosstalk induced jitter model and methods of estimating the contribution of crosstalk to jitter were validated both in simulation and in measurements with good results.

In the second paper, a methodology was proposed using vectorless methods to estimate the statistical characteristics of peak power supply noise and peak jitter due to power supply noise. The mean and standard deviation of the peak power supply noise could be found within 2% and 8%, respectively, using vectorless methods. The mean and standard deviation of peak-to-peak jitter could both be found within 21%. These results are sufficient for determining the impact of a logic design on jitter. This information can be used to guide the development of the logic layout such that its current consumption is within acceptable bounds or to guide the development of defense strategies (e.g. PDN impedance) so that the current consumed by the logic will not generate unacceptable jitter. Statistically meaningful bounds can be placed on the noise or jitter using Chebyshev's inequality, which depends only on the mean and standard deviation.

In the third paper, an analytical delay model of VCDL was proposed to predict propagation delay variations when the power supply is disturbed by an electromagnetic event. Simulated results demonstrate the accuracy of the VCDL delay model. The VCDL analytical delay model is the key part of the overall delay model in DLLs. Similarly, analytical delay models for other DLL components were developed. Then, overall DLL analytical delay model was developed. The proposed analytical delay model is used to predict the output delay variation when the power/ground plan is injected a sinusoidal waveform or EFTs.

VITA

Chunchun Sui was born in Gongzhuling, Jilin Province, P. R. China. He received the Bachelor of Science degree in Mechanical and Electronics Engineering (2005) from Beijing Institute of Technology, Beijing, P.R. China. In 2009, he received the Master of Engineering degree in Software Engineering from Peking University. Beijing, P.R. China. In 2013, he worked as an internship in Hewlett-Packard Company, and Cisco system, San Jose, CA, respectively. In May 2015, he received his Ph.D. in Computer Engineering from Missouri University of Science and Technology.