

DesignCon 2008

A Fast Algorithm to Instantly Predict FPGA SSN for Various I/O Pin Assignments

Geping Liu, Altera Corporation
[gliu@altera.com]

Zhuyuan Liu, Altera Corporation

Kundan Chand, Altera Corporation

San Wong, Altera Corporation

Kaiyu Ren, Altera Corporation

Nafira Daud, Altera Corporation

Abstract

This paper proposes a fast algorithm to instantly predict simultaneous switching noise (SSN) for FPGA I/O pin assignments. In this algorithm, SSN has two distinct components: the mutual inductive coupling and the power distribution network (PDN) noise. Each component can be individually quantified through matrix manipulations and measurement results. This algorithm is independent of any simulator engines, thereby avoiding time-consuming system-level SPICE-like simulations. The predictions from this algorithm are correlated well with direct bench measurements. This paper also describes how to build an SSN measurement automation system to efficiently implement this algorithm, and presents a measurement data interpretation methodology.

Author(s) Biography

Dr. Geping Liu is a supervisory member of technical staff in the characterization group at Altera Corporation. His work and interests include signal integrity, power integrity, and electromagnetic compatibility (EMC) designs in high-speed digital systems, and development and application of numerical and experimental methods in resolving signal integrity and EMC problems. He received his BSEE and MSEE from Tsinghua University, Beijing, China, in 1997 and 1999, respectively, and a Ph.D. in electrical engineering from the University of Missouri-Rolla in 2004.

Zhuyuan Liu is with Altera Corporation as an advanced characterization engineer. She received her BSEE degree from Fudan University, China, and MSEE degree from University of Illinois at Urbana-Champaign. Her interests include signal integrity analysis and high-speed digital system design. She is currently working on characterization, modeling, and correlation of complex FPGAs and printed circuit boards (PCBs).

Kundan Chand received his BS degree in Instrumentation and Electronics Engineering from RV College of Engineering, Bangalore, India in 2002 and MS degree in Electrical Engineering from the University of Missouri-Rolla in 2006. He has worked at Altera Corporation as an advanced characterization engineer since May 2006. His research interests include signal integrity, power integrity, and high-speed digital design. He is currently designing high-speed boards for SSN characterization of Altera FPGAs.

San Wong is a senior manager in the characterization group at Altera Corporation. His current responsibilities include test automation, transceiver characterization, SSN, system-level verification, and other signal integrity solutions. He received his BS in physics from UC Berkeley and MSEE from Stanford University.

Kaiyu Ren received her BS degree in Electrical Engineering from Xi'an Jiaotong University, China in 1997 and MS degree in Electrical Engineering from the University of Toledo in 1998. She has worked at Altera Corporation as a Member of Technical Staff characterization engineer since August 1999. Her research interests include signal integrity and high-speed digital system design.

Nafira Daud is the director of the characterization group at Altera Corporation. He received his BSEE degree from San Francisco State University and MSEE from UC Davis. His responsibilities include device-level characterization and system-level verification infrastructure development, high-speed I/O, memory interface, and power and signal integrity solutions.

I. Introduction

The progress in current CMOS technology, such as advances in 65-nm and 45-nm process, enables a single FPGA to have increased logic element and I/O densities. Meanwhile, lower power supply and higher speed have emerged as principal themes in DDR memory interfaces. Because simultaneous switching noise (SSN) is associated with the number of toggling I/Os, signal voltage level, and switching rate of I/Os, it is important for FPGA users to evaluate SSN performance in a chip-package-board environment.

Recent studies [1], [2] have shown that SSN in an FPGA system is attributed to two primary factors: the mutual inductive coupling among switching I/Os and the impedance profile of a power distribution network (PDN) including die, package and printed circuit board (PCB). In essence, reducing SSN is a design cost issue. While designers can minimize the mutual inductive coupling by increasing the ratio of FPGA ground pins (or return-current pins) to I/O pins, this approach sacrifices I/O densities. Engineers can improve PDN performance by increasing on-die capacitance and add on-package decoupling capacitors, but this approach increases costs. Because FPGAs are programmable, they fit into a wide variety of user applications, so it is useful for designers to use a tool to determine their own SSN budget without additional costs.

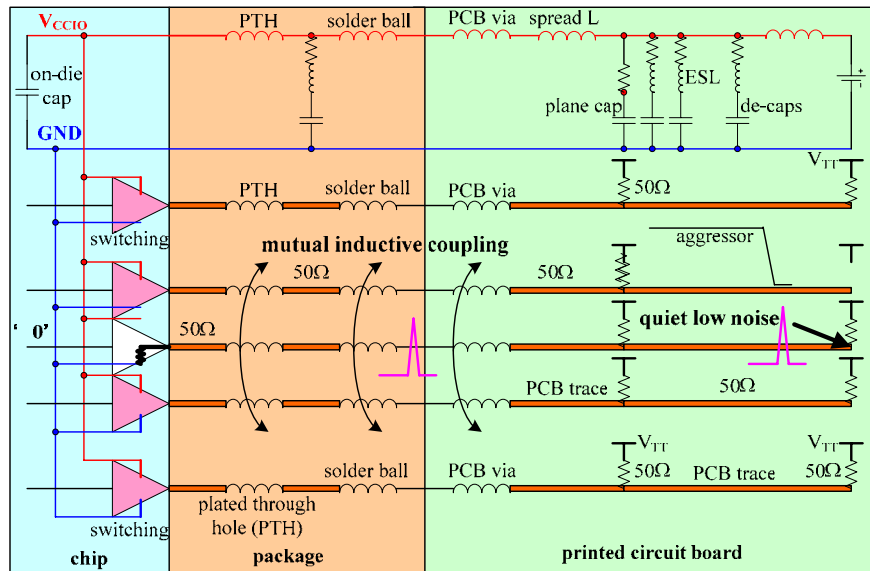
Ideally, this kind of tool requires instant, accurate result predictions for various FPGA I/O pin assignments, and eventually would provide an optimum pin assignment under certain design constraints. In the past, designers have constructed system-level, SPICE-like models to anticipate SSN in FPGA systems. These models are based on an understanding of SSN cause mechanisms, and are correlated well with bench measurements, helping IC and packaging designers improve designs. However, these models are complicated so that signal/power integrity expertise is needed to perform time-consuming system-level simulations. It is cumbersome for all FPGA designers to perform the same level SSN analysis without considering their different design margins. Therefore, this paper describes a fast algorithm to help FPGA users execute a comprehensive FPGA SSN analysis in a short design cycle.

This paper focuses on the SSN caused by FPGA output buffers. This noise is widely known as simultaneous switching output (SSO) noise, and is differentiated from the SSN caused by input buffers. For convenience and consistency, in this paper SSN always refers to the SSO noise. This paper is organized as follows. Section II reiterates two distinct SSN mechanisms in FPGA systems. Section III proposes a measurement-based algorithm to instantly predict the SSN amplitude for an arbitrary FPGA I/O pin assignment. Section IV introduces infrastructure of an SSN automatic measurement system. Section V presents a methodology to interpret SSN measurement data from the automation system. Section VI evaluates the effectiveness of the algorithm as compared to direct bench measurement results. Finally, Section VII provides conclusions and discusses the future work.

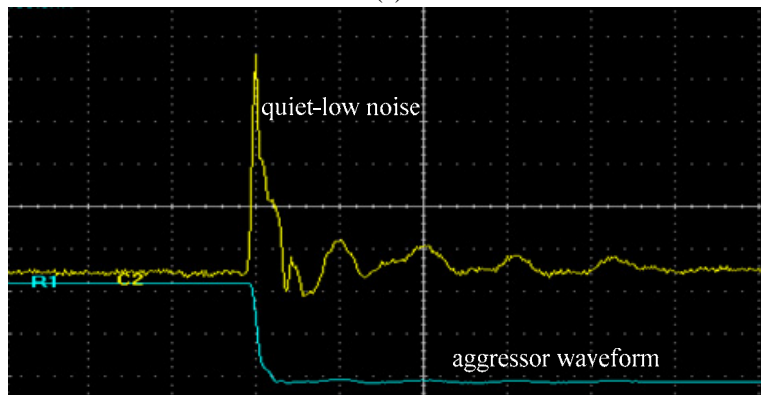
II. SSN Mechanisms in FPGA Systems

The SSN mechanisms in an FPGA system, i.e., the mutual inductive coupling and the delta-I noise in the system-level PDN, have been widely understood. Moreover, the contributions from the inductive coupling and the PDN effect can be decomposed and individually quantified.

The scenario of quiet-low noise is illustrated in Figure 1. A quiet pin, i.e., a victim pin, is held at logic low, and forms a closed loop with its return current path. Similarly, its adjacent I/Os, i.e., aggressor pins, form closed loops with their return current paths. When multiple I/Os switch simultaneously, the transient currents through them create time-varying magnetic fields, which penetrate into the victim loop and induce a voltage noise. A typical quiet-low noise waveform is shown in Figure 1 (b). In this case, the mutual inductive coupling is the dominant factor.



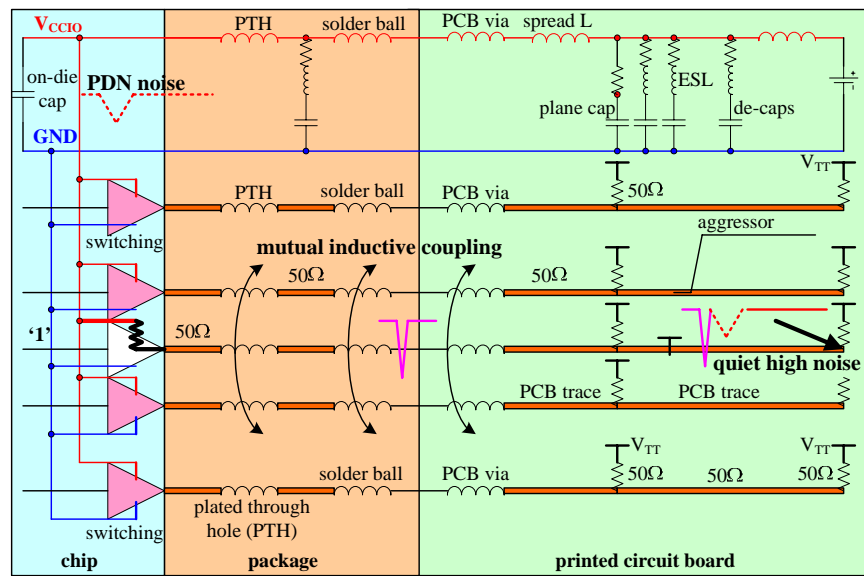
(a)



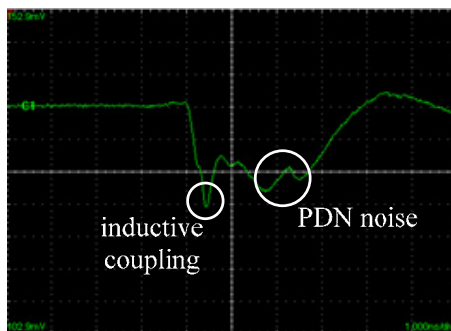
(b)

Figure 1. Quiet-low noise scenario: (a) Schematic of an FPGA system at aggressors' falling edge; (b) Measured quiet-low noise waveform at a victim pin

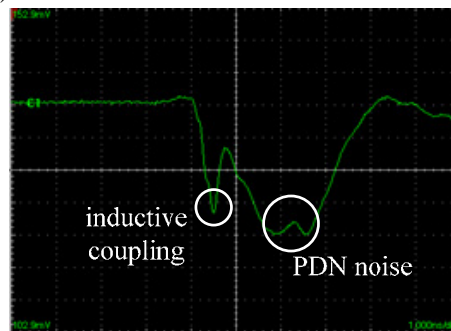
The quiet-high noise scenario is illustrated in Figure 2. A quiet pin is held at logic high, and a number of output buffers switch at the same time. A substantial number of transient currents not only generates magnetic fields coupled to the victim pin, but also causes a delta-I voltage drop when it flows through the inductive circuit components of the PDN. These inductances are associated with interconnection structures, such as power balls of a ball-grid array (BGA) package and power vias in a PCB. The contributions from the mutual inductive coupling and the PDN effect are distinct, as shown in Figure 2 (b) and (c). The sag caused by the mutual inductive coupling has a higher resonant frequency, as compared to the sag due to the PDN effect. This important characteristic facilitates the decomposition of the two contributions. When only a few of aggressors toggle, the mutual inductive coupling usually contributes more than the PDN effect does, as shown in Figure 2 (b). However, the PDN sag might be worse when more aggressors switch simultaneously, as shown in Figure 2 (c).



(a)



(b)



(c)

Figure 2. Quiet-high noise scenario: (a) Schematic diagram of an FPGA system at aggressors' rising edge; (b) Measured quiet-high noise waveform when the inductive coupling is worse; (c) Measured quiet-high noise waveform when the PDN noise is worse

III. Algorithm for Instant SSN Prediction

Because the SSN mechanisms have been well understood, a fast algorithm is proposed to instantly predict amplitude of SSN glitch to avoid any SPICE-like simulations. A high-level flow chart is shown in Figure 3. The input of this algorithm is an FPGA I/O pin assignment, where an I/O standard, pin count, and pin locations need to be specified. The output of this algorithm is the glitch amplitude of quiet-high or quiet-low noise. In Step 1 of this flow chart, the mutual inductive coupling is quantified for both quiet-high and quiet-low case. In Step 2, the PDN noise is characterized for quiet-high case only. In Step 3, the mutual inductive coupling is reported as the quiet-low noise. In Step 4, the worse glitch number between Step 1 and 2 is reported as the quiet-high noise.

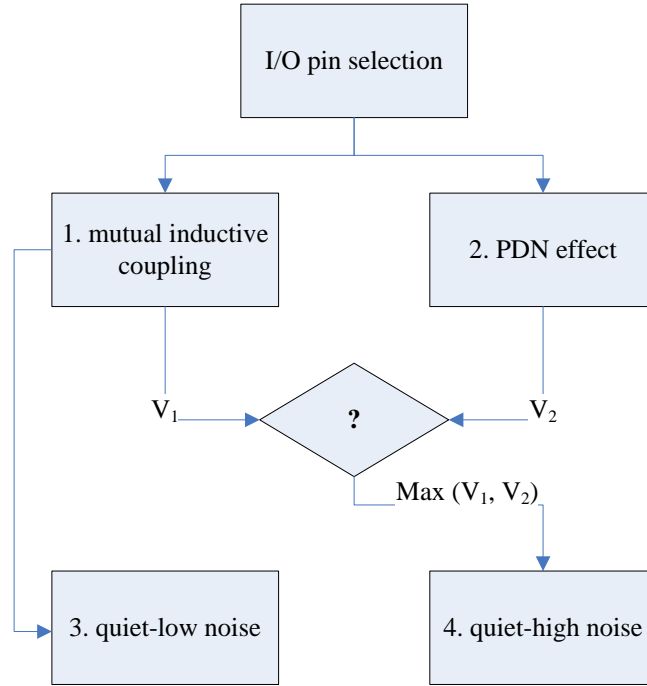


Figure 3. High-Level Flow Chart of this Fast SSN Prediction Algorithm

Mutual Inductive Coupling Matrix in Step 1

The mutual inductive coupling only happens when current changes as a function of time, and the governing equation is described as:

$$V_{ij} = M_{ij} \frac{dI_j}{dt}, \quad (1)$$

where M_{ij} is the mutual inductance between a quiet pin i and an aggressor pin j , and I_j is the current flowing through pin j . When more aggressors toggle, the noise at the quiet pin i can be written as:

$$V_i = \sum_{j=1}^N M_{ij} \frac{dI_j}{dt}, \quad (2)$$

where N is the number of aggressors. A matrix manipulation can be applied to generalize this superimposition principle in (2) for a whole FPGA package:

$$\begin{array}{ccc}
 \begin{array}{c} \text{vector of} \\ \text{victim pins} \end{array} & & \begin{array}{c} \text{matrix of mutual} \\ \text{inductive coupling} \end{array} & & \begin{array}{c} \text{vector of} \\ \text{aggressor pins} \end{array} \\
 \left(\begin{array}{c} V_1 \\ \vdots \\ V_N \end{array} \right) & = & \left(\begin{array}{ccc} & & \\ & & V_{ij} \\ & & \end{array} \right) & & \left(\begin{array}{c} 0 \\ 1 \\ 1 \\ \vdots \\ 1 \\ 0 \\ 0 \\ 1 \end{array} \right) \\
 N \times 1 & & N \times N & & N \times 1
 \end{array} \quad (3)$$

where N is the total I/O number of a certain package. For each toggling aggressor pin, the corresponding element is defined as ‘1’ in the vector of aggressor pins in (3), while a non-toggling pin is specified as ‘0’ instead. The element V_{ij} ($i, j = 1, 2, \dots, N$) in this matrix is the voltage coupled to a quiet victim pin i when an aggressor pin j is toggling. All these elements can be extracted from an automatic measurement system, which is introduced in Section IV. The property of this matrix in (3) is summarized as follows:

- It is a symmetrical matrix, i.e., $V_{ij} = V_{ji}$ ($i, j = 1, 2, \dots, N$)
- For all diagonal elements, $V_{ii} = 0$, $i = 1, 2, \dots, N$

In reality, the value of V_{ij} is close to zero, when pin i and pin j are located far away from each other. Therefore, the mutual inductive coupling matrix is sparse in nature.

Correction Factor Curve in Step 1

When more than one aggressor switches, equation (3) usually overestimates the mutual inductive coupling as compared to direct bench measurements, as illustrated in Figure 4. This reveals an important SSN mechanism that buffer switching rate degrades with the total number of aggressors and relative locations of neighboring buffers. When more aggressors toggle on the same time, the on-chip PDN may not be sufficient to provide enough charges in a short time, increasing the transition time of drivers. When two switching aggressors are closely located, the mutual inductance between them also slows down switching process of the two drivers. In addition, the non-linear nature of I/O output buffer increases the complexity of analysis.

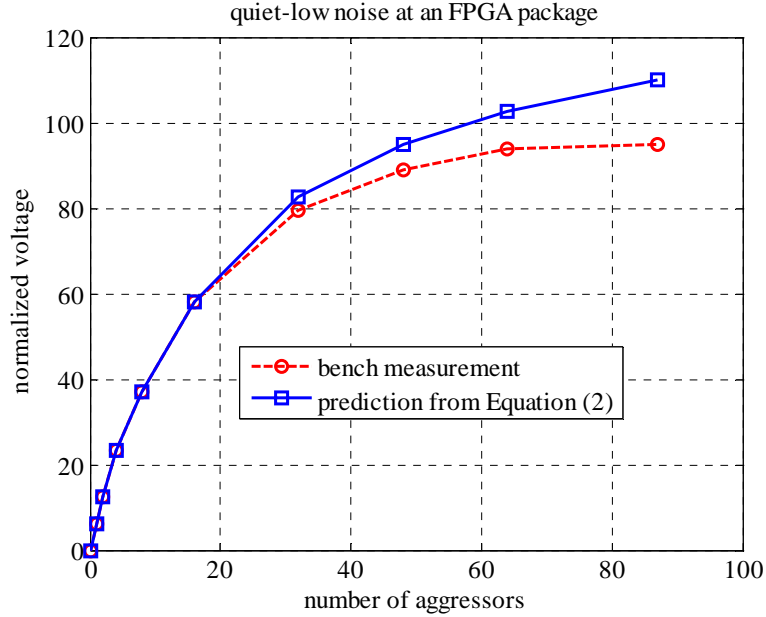


Figure 4. Quiet-low noise comparison between direct bench measurements and predictions from equation (2) or (3). (A victim pin is specified, and aggressors are selected in a binary and outward spiral pattern starting from the vicinity of this victim pin.)

A correction factor curve can be extracted from bench measurements, and applied in (3) to improve its accuracy. If the switching rate of a single aggressor in (1) is defined as a reference, the switching rate of multiple aggressors can be normalized, and the value should be ≤ 1 . Figure 5 shows a normalized correction factor curve for a wire-bond packaged FPGA. For simplicity, this curve represents an average switching behavior of buffers. In this fast algorithm, all toggling aggressors are sequentially analyzed, and appropriate correction factor values are assigned to aggressors accordingly. Therefore, the simple equation (3) is upgraded to (4). The mutual inductive coupling matrix remains the same as compared to that in (3). Each 0 in the corrected aggressor vector in (4) still represent non-active, tri-state output buffers. s_i is the correction factor of a switching aggressor pin, where $i \in 1, 2, \dots, N$.

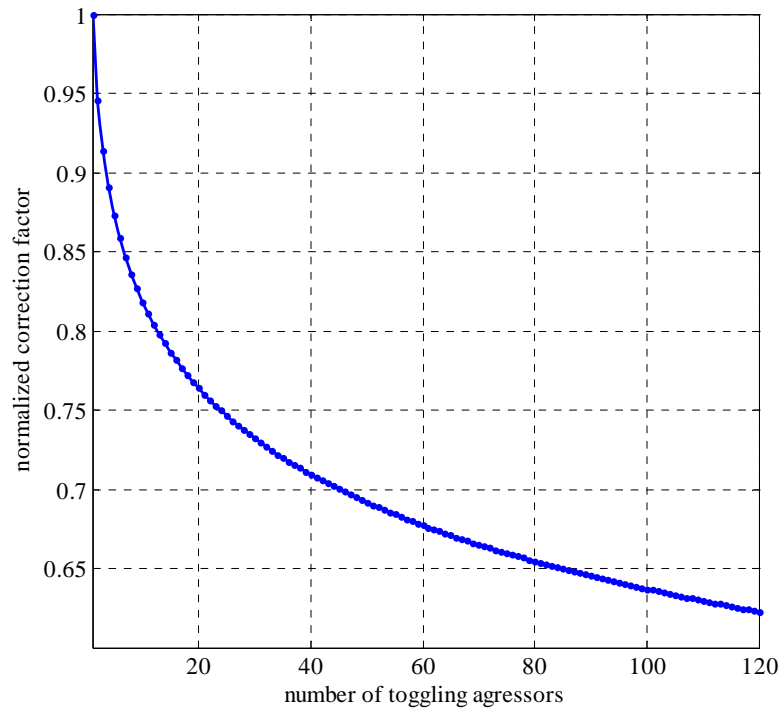


Figure 5. Correction factor curve for toggling aggressors

$$\begin{array}{c}
 \begin{array}{|c|} \hline \text{victim vector} \\ \hline \end{array} \\
 \begin{pmatrix} V_1 \\ \vdots \\ V_N \end{pmatrix} \\
 N \times 1
 \end{array}
 =
 \begin{array}{c}
 \begin{array}{|c|} \hline \text{mutual inductive coupling} \\ \text{matrix} \\ \hline \end{array} \\
 \begin{pmatrix} & & & & & \\ & & & & & \\ & & & & & \\ & & & V_{ij} & & \\ & & & & & \\ & & & & & \\ & & & & & \end{pmatrix} \\
 N \times N
 \end{array}
 \begin{array}{c}
 \begin{array}{|c|} \hline \text{corrected} \\ \text{aggressor vector} \\ \hline \end{array} \\
 \begin{pmatrix} 0 \\ s_2 \\ s_3 \\ \vdots \\ 0 \\ 0 \\ s_N \end{pmatrix} \\
 N \times 1
 \end{array}
 \tag{4}$$

Scaling Factor from One Matrix to the Other in Step 1

When a quiet-high mutual inductive coupling matrix is extracted for a certain type of I/O standard, it can be easily scaled by using a single constant to create a quiet-low matrix for that I/O standard. For the same I/O standard, the relationship between the PMOS DC resistance and the NMOS DC resistance is fixed, and the package connection and PCB trace are also identical. Therefore, the ratio of each element in a quiet-high matrix to the corresponding element in the quiet-low matrix is approximately constant. This analysis

has been verified with bench measurements, as shown in Figure 6. In this case, a quiet-high matrix is measured for a wire-bond packaged FPGA in SSTL1.8V Class II (16 mA current strength setting). A constant of 0.95 is successfully used to scale the quiet-high matrix to generate a quiet-low matrix.

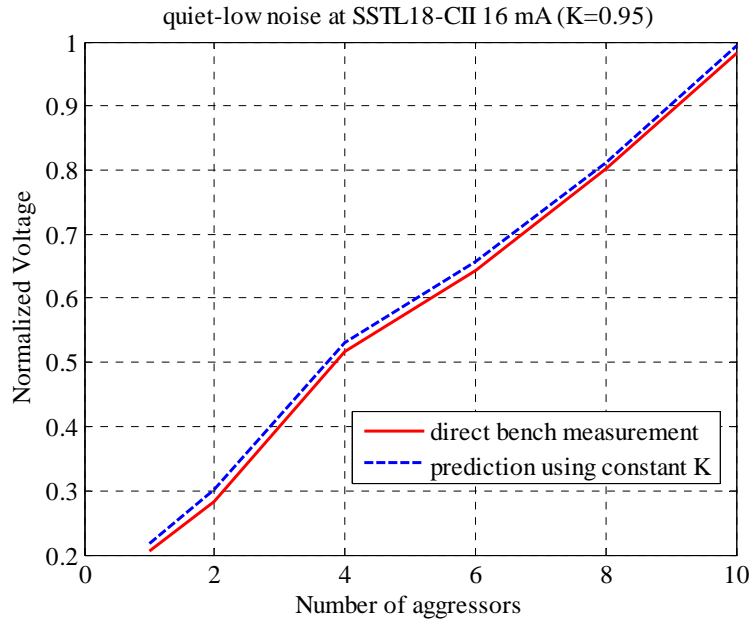


Figure 6. Scaling a quiet-high mutual inductive coupling matrix to a quiet-low matrix by using a single constant of 0.95

Similarly, the quiet-high matrix of an I/O standard can be scaled to obtain both quiet-high and quiet-low matrices of another I/O standard, as long as the two I/O standards have the same termination topology on the package and PCB. In general, there are three categories of trace terminations:

- Non-voltage referenced LVTTTL/LVCMOS standards;
- HSTL/SSTL Class I standards with a single parallel termination resistor connected to V_{TT} ;
- HSTL/SSTL Class II standards with two parallel termination resistors connected to V_{TT} .

In Figure 7, a quiet-high matrix for SSTL1.8V Class II, 25Ω on-chip-termination (OCT) is known. When it is multiplied with a scaling factor of 1.44 and 1.40, the quiet-high and quiet-low matrices of a new I/O standard, SSTL1.8V Class II (16-mA current strength), can be generated, respectively. The two I/O standards belong to the same third termination category.

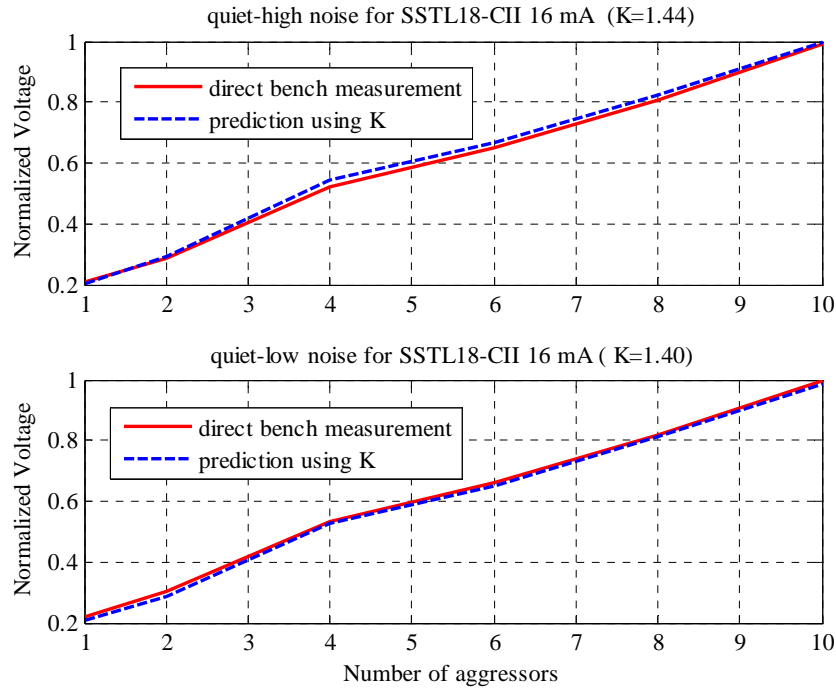


Figure 7. Scaling a quiet-high matrix of SSSL1.8V Class II (25Ω OCT) to generate a quiet-high and a quiet-low matrix for SSSL1.8V Class II (16-mA current strength)

PDN Effect in Step 2

For a typical quiet-high noise waveform shown in Figure 2, the relationship of the PDN sag amplitude to the number of aggressors can be quantified with bench measurements. In general, the PDN noise amplitude monotonically goes up as the aggressor number increases, as illustrated in Figure 8. This information is compared with the contribution from the mutual inductive coupling in Step 1, and the worse one is reported as a quiet-high noise in Step 4.

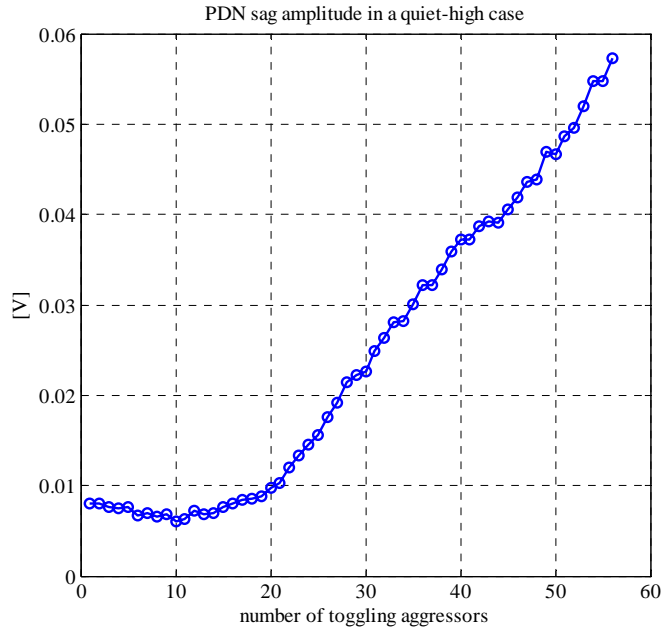


Figure 8. PDN noise amplitude versus the number of toggling aggressors in a quiet-high noise case

IV. Automatic SSN Measurement System

An automatic SSN measurement system has been developed to efficiently implement the fast algorithm in this paper. The system uses both software and hardware. The software system includes a Nios[®] II embedded processor in the FPGA device under test (DUT), a Nios II processor in the control FPGA, and an integrated characterization environment (ICE) running on the PC. The key hardware consists of broadband multiplexers.

The ICE program, which controls the entire automation system, has three main functions

- Controls all test equipment, such as DC power supplies, a pulse generator, and a high-bandwidth sampling oscilloscope
- Sends defined noise patterns to the Nios II processor embedded in the DUT
- Sends defined commands to the Nios II processor in the control FPGA to configure all multiplexers

Because the Nios II processor is a soft core, it can establish a customized protocol to communicate with the PC. In this protocol, a basic element is an SSN pattern including the selection of a victim pin and multiple aggressor pins. Therefore, a pattern or a series of patterns can be conveniently defined in the PC. Then these patterns are sent to the Nios II processors in the DUT and the control FPGA, respectively, through an RS232 series interface. The Nios II processor dynamically programs I/O pins of the DUT. Meanwhile, the Nios II processor in the control FPGA can dynamically control the multiplexers so that the SSN waveform at the victim pin can be captured with the sampling oscilloscope.

The automation system requires broadband multiplexers to select a victim pin among all FPGA I/Os. Currently, this multiplexing structure can support up to 800 I/Os, and has a bandwidth from DC to 8 GHz. In addition, a 450Ω resistor is added between the far-end of each I/O trace and each channel of multiplexers, and it forms a 10:1 probe with a 50Ω oscilloscope termination, as shown in Figure 9.

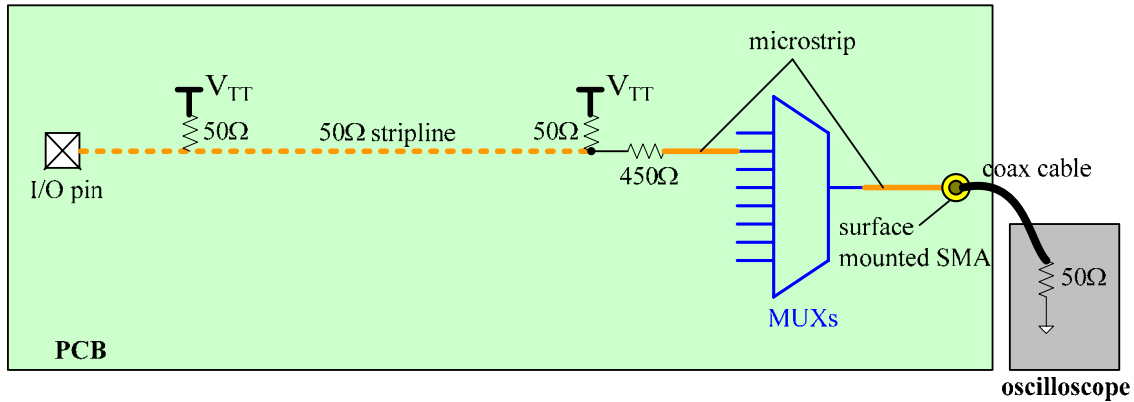


Figure 9. Typical I/O trace termination topology with broadband multiplexers

V. Interpretation of SSN Measurement Data

Based on the mechanism analysis in Section II, the scenario of quiet-high or quiet-low noise can be considered to have a lumped voltage source attached to the victim pin at the package/PCB breakout area. The first incident SSN glitch waveform travels along the victim trace on the board, and reaches the far end, as shown in Figure 10. The characterization board is well-designed so that the inductive coupling between the victim trace and the aggressor traces are negligible during the wave propagation process. Therefore, quantifying this incident SSN waveform along the trace is a good measure of SSN noise source characteristics.

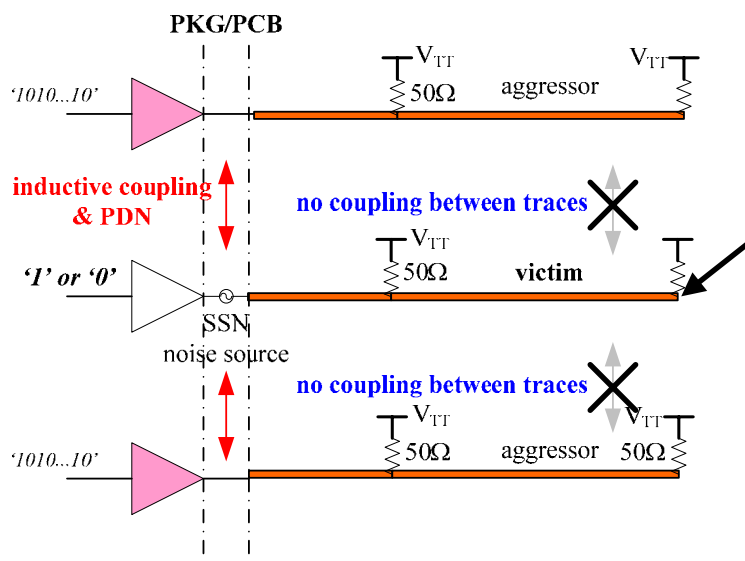


Figure 10. Simplified diagram of generation and transmission of an incident SSN glitch waveform

The 10:1 on-board passive probe is used to measure the quiet-high and quiet-low noise at the far end of I/O traces. The 500Ω input impedance degrades the measurement fidelity, i.e., it affects the SSN glitch amplitude. Therefore, the probe effect needs to be de-embedded for extracting the incident SSN waveform. For the voltage referenced I/O standards shown in Figure 11, such as SSTL Class I & II and HSTL Class I & II, the reflection coefficient Γ and the transmission coefficient T can be calculated as:

$$\Gamma = \frac{Z_L - Z_0}{Z_L + Z_0} = \frac{500 // 50 - 50}{500 // 50 + 50} = -0.0476, \quad (5)$$

$$T = 1 + \Gamma = 0.9524. \quad (6)$$

The incident SSN glitch without any probe loading can be written as:

$$V_2 = V_1 * 10 / T. \quad (7)$$

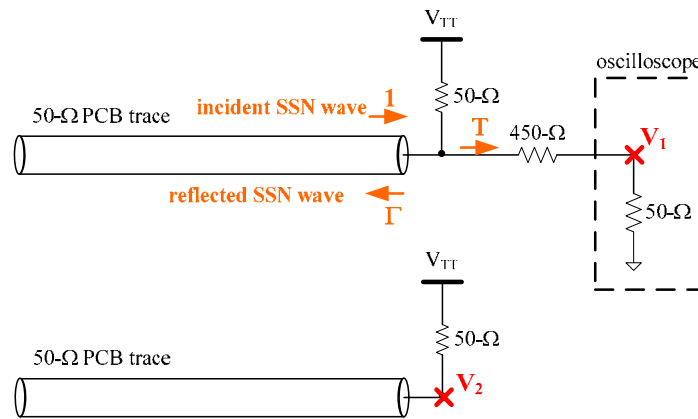


Figure 11. De-embedding the 10:1 passive probe effect for voltage-referenced I/O standards, such as SSTL Class I & II and HSTL Class I & II.

Similarly, for the non-voltage referenced I/O standards shown in Figure 12, such as LVTTTL and LVCMOS, the reflection coefficient Γ is about 0.818 and the transmission coefficient T is about 1.818. The incident SSN glitch wave can be reconstructed accordingly [3].

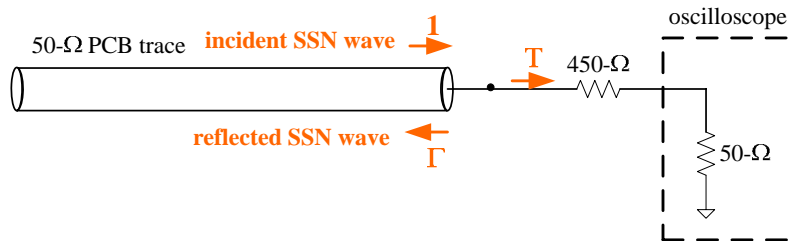


Figure 12. De-embedding the 10:1 passive probe effect for non-voltage referenced I/O standards, such as LVTTTL and LVCMOS

The incident SSN waveform $V_s(t)$ along a 50Ω trace can be used to anticipate the worst SSN glitch amplitude when a loading capacitor is added at the far end of the victim trace, as illustrated in Figure 13. A voltage source in series with a 50Ω source impedance is

constructed, and the amplitude is two times $V_s(t)$. This voltage source is transformed to the frequency domain through an FFT algorithm, and multiplied by a frequency-domain transfer function $G(f)$, which consists of the loading capacitor. Then the multiplication result is transformed back to the time-domain waveform $V_L(t)$.

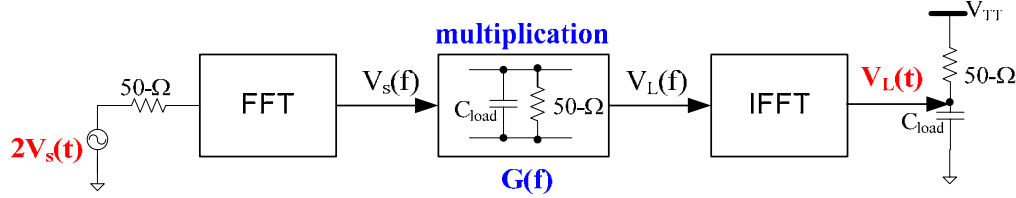


Figure 13. Schematic of SSN glitch prediction at a loading capacitor, when the incident SSN waveform along a 50Ω trace is given

An FPGA with a wire-bond package is used to verify this analytical approach. When an SSTL1.8V Class II (25Ω OCT current strength setting) is used and 26 aggressors toggle, the quiet-high and quiet-low noise waveforms at a 5 pF loading capacitor are shown in Figure 14. The SSN glitch prediction results are correlated well with HSPICE simulation and direct bench measurement results.

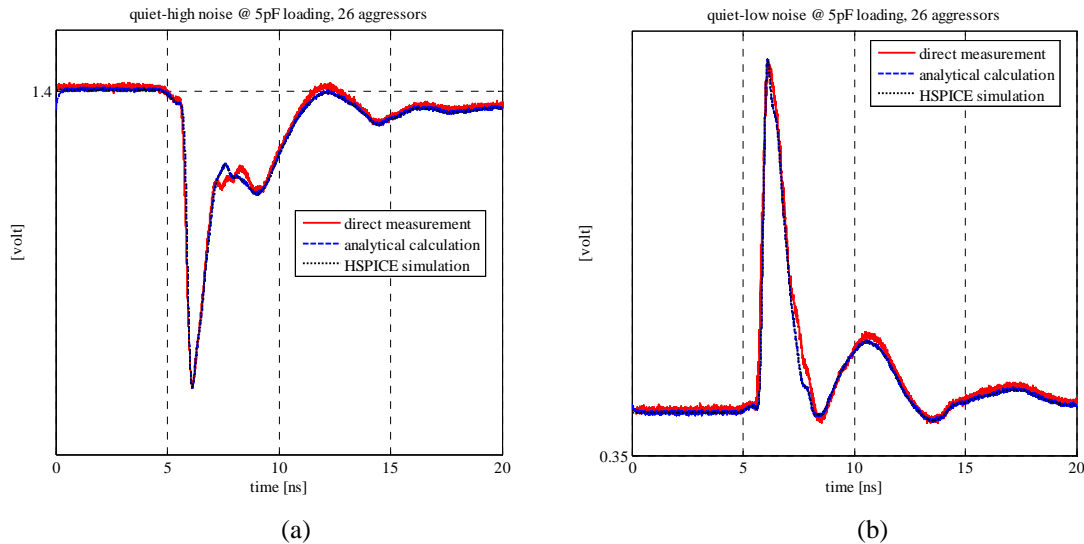


Figure 14. Comparison of the analytical calculation of SSN glitch at a 5-pF loading capacitor with direct bench measurement and HSPICE simulation results (SSTL1.8V Class II, 25Ω OCT current strength setting, 26 aggressors); (a) Quiet-high noise at the 5-pF loading capacitor; (b) Quiet-low noise at the 5-pF loading capacitor

VI. Correlations to Direct Measurements

A 324-pin wire-bond FPGA in a BGA package with a supporting board is used to evaluate the fast algorithm in section III. The 24-layer board is $12'' \times 10'' \times 0.13''$ in size, and the dielectric material is G-tek with a relative permittivity of 4.0. This board has six inner signal layers for all I/O traces and three power planes for supplying the DUT. This DUT has 174 I/O pins bonded out on the board. Each I/O trace is a ground-referenced,

single-ended 50Ω stripline, and is 5 inches long. For this wire-bond package, package-level mutual inductive coupling is the dominant SSN mechanism, and its amplitude is reported as either a quiet-high or a quiet-low noise.

A quiet-high noise case is investigated with an I/O standard of SSTL2.5V Class II (25Ω OCT current strength setting). The trace termination topology can be referred to Figure 9. No loading capacitor is at the far-end of PCB traces. First, the automatic SSN measurement system is applied to extract a quiet-high, mutual inductive coupling matrix for all 174 I/Os. Second, the correction factor curve is quantified with measurements. Finally, equation (4) is fully established to instantly predict SSN number for an arbitrary I/O pin assignment.

A statistical verification methodology is used in this paper. 1,720 types of quiet-high noise patterns are generated randomly. A quiet-high noise pattern is a Quartus® II I/O pin assignment including a quiet-high victim pin and multiple toggling aggressor pins. The number of aggressors, the location of aggressors, and the location of a victim pin are all randomly chosen. A quiet-high noise distribution of these random bench measurements is shown in Figure 15. The error distribution across the patterns between equation (4) and these direction bench measurements is shown in Figure 16. The error bound is within ±18%, and most of predictions have less than ±10% difference. Moreover, the error percentage decreases to ±5% for cases with higher SSN glitch amplitudes, as shown in Figure 17.

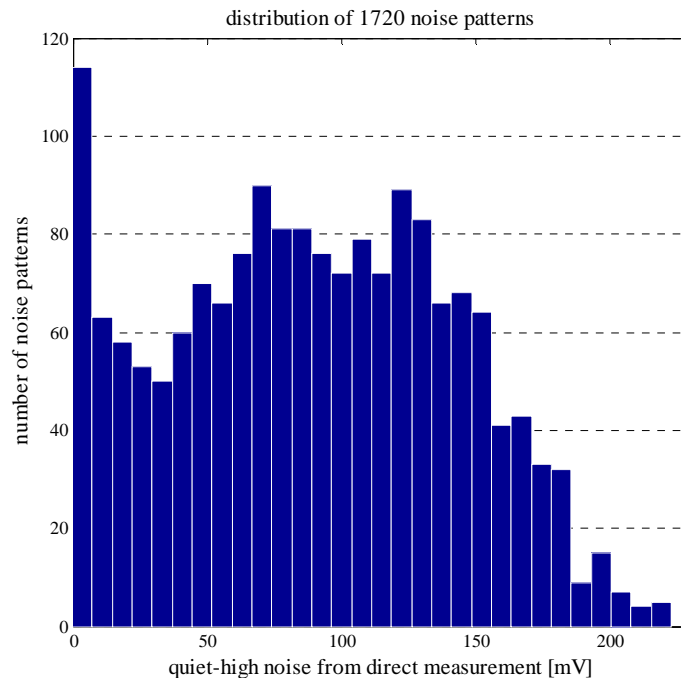


Figure 15. Quiet-high noise distribution of 1,720 random patterns for an I/O standard of SSTL 2.5V Class II (25Ω OCT current strength) without any loading capacitors

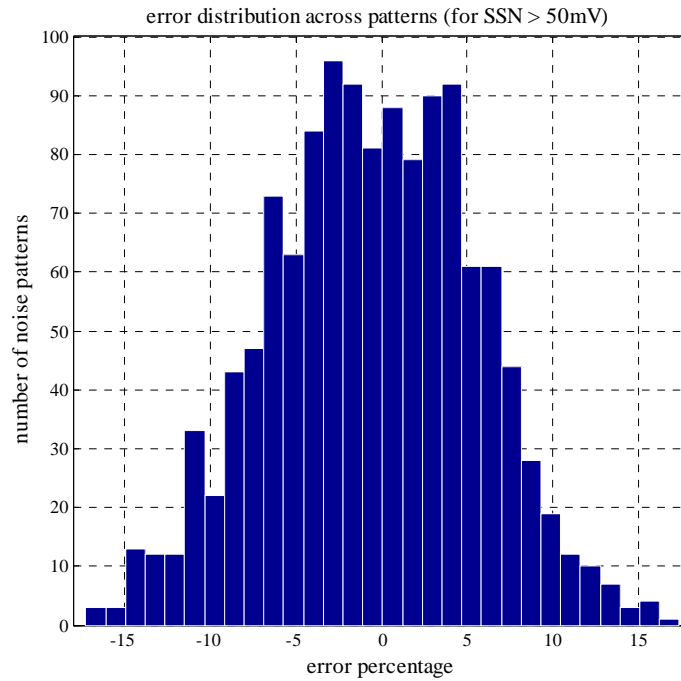


Figure 16. Error distribution between the predictions from Equation (4) and direct bench measurements across the 1,720 random noise patterns, with I/O standard SSTL 2.5V Class II (25Ω OCT current strength) without any loading capacitors

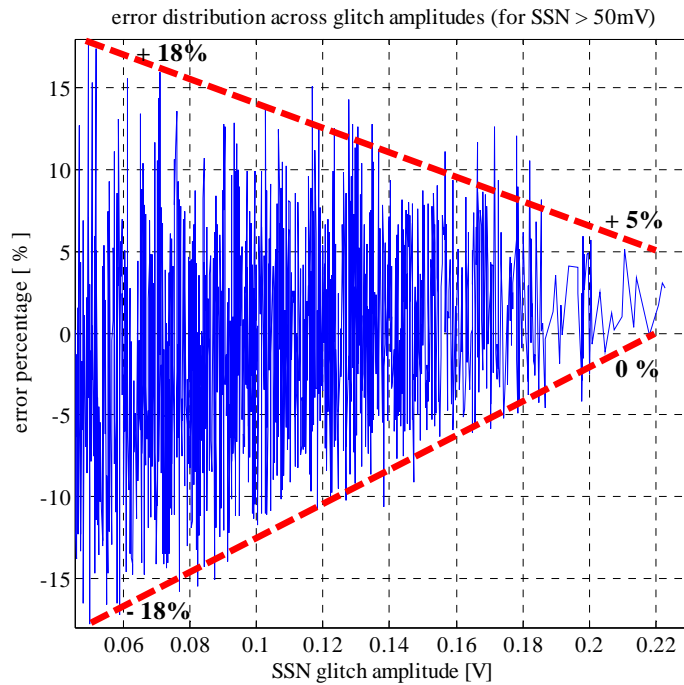


Figure 17. Error distribution between the predictions from Equation (4) and direct bench measurements across the SSN glitch amplitudes of 1,720 random patterns, with I/O standard SSTL 2.5V Class II (25Ω OCT current strength) without any loading capacitors

VII. Conclusions and Future Work

In this paper, the two distinct SSN mechanisms—the mutual inductive coupling between I/Os and the PDN effect—are iterated. Moreover, the two contributions can be decomposed and quantified individually. In the quiet-low noise case in Figure 1, the mutual inductive coupling is always the primary factor. In the quiet-high noise case in Figure 2, the mutual inductive coupling is still dominant for wire-bond packaged FPGAs. However, it is not always true for FPGAs with flip-chip packages. With an increased switching I/O number, the PDN effect may contribute more than inductive coupling.

A measurement-based analytical equation (4) is proposed to instantly predict the quiet-high or quiet-low noise for an arbitrary FPGA I/O pin assignment. To quantify the mutual inductive coupling, an I/O pin-aware matrix can be extracted from an automatic measurement system. Each element in this matrix represents the inductive coupling between a victim pin and an aggressor pin. For designs with multiple aggressor pins, the superimposition theory can be applied to add up the individual contributions with appropriate correction factors for the aggressors. Meanwhile, the relationship of PDN noise to the number of aggressors can be quantified with bench measurement, as shown in Figure 8. A flow chart shown in Figure 3 is used to execute this fast algorithm. The predictions of SSN glitch amplitude are correlated well with direct bench measurements for FPGAs in wire-bond packages. The error bound over 1,720 quiet-high random noise patterns is within $\pm 18\%$, as shown in Figure 16 and Figure 17.

In practice, the implementation of the fast algorithm relies on a measurement automation system, which is described in Section IV. In this automation system, an ICE program running in a PC controls the automation flow and external equipments, and communicates with Nios II soft processors in the DUT and the control FPGA. The DUT can be programmed accordingly. The control FPGA executes necessary logic controls over broadband multiplexers so that the noise waveform at the victim pin can be captured with a high-bandwidth sampling oscilloscope.

Because on-board 10:1 passive probes are widely used in the automation PCB, an SSN data interpretation methodology is applied to de-embed the probe effect, and the characteristics of incident SSN glitch along a 50Ω trace of victim pin are characterized. Moreover, the loading capacitor effect on SSN glitch can be predicted well by using an analytical approach in Figure 13.

In the future, the current algorithm can be improved in several aspects. To increase accuracy, some missing second- or third-order effects need to be considered in the correction factor curve. For FPGAs in flip-chip packages, the effects of the PCB need to be de-embedded. Furthermore, this algorithm should be capable of incorporating the parameters of customers' boards to anticipate the corresponding SSN glitch amplitude.

References

- [1] H. Shi, G. Liu, and A. Liu, "Analysis of FPGA simultaneous switching noise in three domains: time, frequency, and spectrum", *Proc. DesignCon 2006*, Feb. 2006.
- [2] G. Liu, H. Shi, S. Wong, and A. Chang, "Analyzing FPGA simultaneous switching noise in printed circuit boards", *IEEE EMC Symposium*, Portland, 2006.
- [3] W. Lo, L. Smith, G. Liu, M. Wong, N. Daud, "Theoretical basis and measurement technique for SSN (simultaneous switch noise) on FPGA (field programmable gate array) Products", accepted by *IEEE APACE 2007*.

Acknowledgements

The authors would like to thank Weiwei Lo, Larry Smith, John Oh, and Hong Shi for fruitful discussions with them about the ongoing SSN investigation, and the IC Design, Product Engineering, Application Engineering, and Packaging Engineering departments for supporting this work.



101 Innovation Drive
San Jose, CA 95134
www.altera.com

Copyright © 2008 Altera Corporation. All rights reserved. Altera, The Programmable Solutions Company, the stylized Altera logo, specific device designations, and all other words and logos that are identified as trademarks and/or service marks are, unless noted otherwise, the trademarks and service marks of Altera Corporation in the U.S. and other countries. All other product or service names are the property of their respective holders. Altera products are protected under numerous U.S. and foreign patents and pending applications, maskwork rights, and copyrights. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera Corporation. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.