MODELLING OF CROSSTALK NOISE IN LOCAL INTERCONNECTS

A DISSERTATION

Submitted in partial fulfillment of the requirements for the award of the degree of

MASTER OF TECHNOLOGY

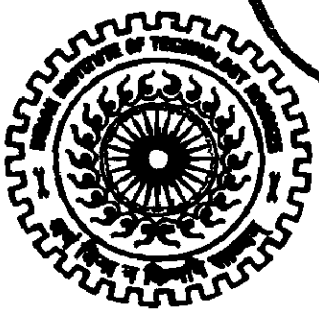
in

ELECTRONICS AND COMPUTER ENGINEERING

(With Specialization in Semiconductor Devices & VLSI Technology)

By

TAPAS CHOUDHURY



DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE -247 667 (INDIA)
JUNE, 2009

CANDIDATE'S DECLARATION

I hereby declare that the work, which is presented in this project report entitled, **"Modelling of Crosstalk Noise in Local Interconnects"**, being submitted in partial fulfillment of the requirements for the award of degree of MASTER OF TECHNOLOGY with specialization in Semiconductor Devices & VLSI Technology (SDVT), in the Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, under the supervision of Dr. S. Dasgupta, Assistant Professor, Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, has been written by me and only I am /I will be responsible for it. I have not submitted the record embodied in this report for the award of any other degree or diploma.

Tapas

Date: 23/06/09
Place: Roorkee

(TAPAS CHOUDHURY)
M.Tech II Year (SDVT)
E&C Department,
IIT Roorkee.

This is to certify that the above statement made by the candidate is correct to the best of my knowledge and belief.

Date: 23/06/09
Place: Roorkee

Dr. S. DASGUPTA
Assistant Professor, E&C Dept.
IIT Roorkee,
Roorkee – 247 667, INDIA.

ACKNOWLEDGEMENT

At the outset, I express my heartfelt gratitude to Dr. S. Dasgupta, Asst. Professor, Department of Electronics and Computer Engineering at Indian Institute of Technology, Roorkee for his valuable guidance, support, encouragement and immense help. I consider myself extremely fortunate for getting the opportunity to learn and work under his able supervision. I have deep sense of admiration for his innate goodness and inexhaustible enthusiasm. It helped me to work in right direction to attain desired objectives. Working under his guidance will always remain a cherished experience in my memory and I will adore him throughout my life.

My sincere thanks are also due to rest of the faculty in the Department of Electronics and Computer Engineering at Indian Institute of Technology Roorkee for the technical knowhow and analytical abilities they have imbibed in us which have helped me in dealing with the problems I encountered during the dissertation.

I am greatly indebted to all my friends, who have graciously applied themselves to the task of helping me with ample moral support and valuable suggestions. Finally, I would like to extend my gratitude to all those persons who directly or indirectly helped me in the process and contributed towards this work.

Tapas Choudhury
M.Tech (SDVT)

Abstract

Process variations and crosstalk noise are two seemingly uncorrelated issues that affect performance in Ultra Deep Submicron designs. Both start to play a crucial role with decreasing process nodes and increasing design density. In addition to this, crosstalk noise has an indirect dependence on process variations. A simple framework is proposed for doing statistical analysis of the effect of crosstalk noise on the functionality of logic gates. The dependence of functional noise on process variations is looked into by analyzing the variation of both glitch peak and area with process parameters. The effect of circuit and device parameters on delay noise and propagated noise is discussed. Thus the theoretical framework for statistical characterization of logic gates for noise rejection is laid down by proposing a statistical representation of the Noise Rejection Curves. The proposed framework is validated using parametric simulations in 90nm technology.

Contents

List of figures

1. Crosstalk noise

1.1 Introduction

Crosstalk noise is defined as the change in the voltage waveform of a net in an undesired way, due to the signal activity in its neighbouring nets, which are capacitively coupled to it [1]. Crosstalk noise has become a critical design and verification issue for large, high-performance designs. In noise analysis, the nets on which crosstalk noise is injected by one or more of its neighbors are called the victim nets, whereas the nets that inject this noise are called the aggressor nets.

Crosstalk noise can manifest itself in two ways. Functional noise refers to noise that occurs on a victim net which is held quiet by a driver. Crosstalk noise on such a victim causes a glitch (Fig. 1) which may propagate to a dynamic node or a latch, changing the circuit state and causing a functional failure. On the other hand, delay noise refers to noise that occurs when two capacitively coupled nets switch simultaneously (Fig. 1). Depending on the direction of these transitions, the delays on both nets are affected giving rise to potential setup or hold time failures.

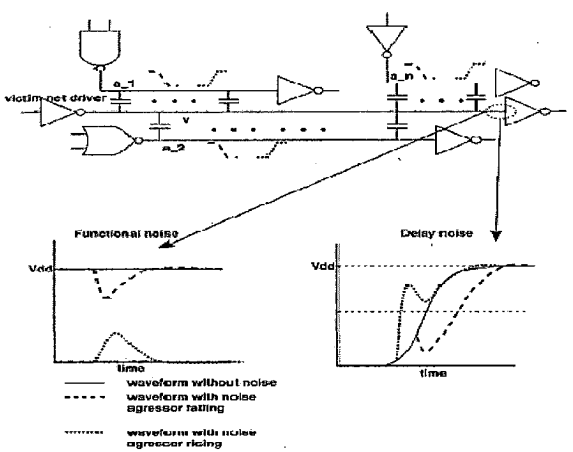


Figure 1 Effects of crosstalk noise [1]

1.2 Growing importance of crosstalk noise

Although crosstalk noise has always existed in integrated circuits, it has become a critical issue due to the following reasons [2][3]. Finer geometries and increasing interconnect density along with more metal layers have resulted in greater wire and via resistances. Narrow wires have also become thicker to cope with increased resistivity, thus resulting in an increased wire aspect ratio, which translates into an increased ratio of crosstalk capacitance to total capacitance. On the other hand, the usage of more aggressive and less noise immune circuit structures such as dynamic logic has increased due to performance reasons.

Lower device lengths have resulted in faster but low threshold gates. Together with lower supply voltages, the noise margins of these high-performance gates have been significantly lowered. Faster slews have resulted in increased injected noise whereas smaller clock cycles dictate much less tolerance to delay variations.

Aggressive VLSI design methodologies lead to increased system performance variation [4][5]. For example, reduced supply voltage and transistor threshold voltage imply reduced noise margin and increased variability; increased device density in a single chip results in increased supply voltage and temperature variations; higher operating frequencies lead to increased capacitive and inductive couplings on silicon surface and in silicon substrate; aggressive performance optimization increases the number of near-critical paths and the probability of timing failure.

With the advance of process technology, cross-coupled noise in digital designs has greatly increased especially in high performance designs that employ fast signal transition times. This trend has led to the need for accurate noise analysis tools [6][7]. A victim net with its associated aggressor nets is referred to as a noise cluster. Typically, two types of noise are distinguished. Functional noise occurs when the victim net is in a stable state and the aggressor nets switch. In this case, a noise pulse occurs on the victim net that can change the state of a storage element, such as a latch, and can cause a functional failure. Delay noise occurs when the victim net transitions simultaneously with the aggressor nets that inject noise upon it. In this case, the delay of the net is modified leading to performance violations.

inject noise upon it. In this case, the delay of the net is modified leading to performance violations.

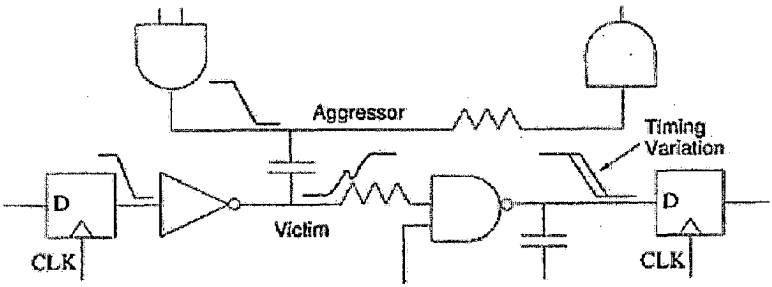


Figure 2 Crosstalk inducing timing variation [25]

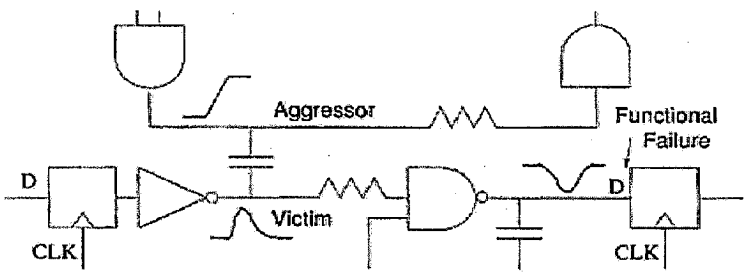


Figure 3 Crosstalk inducing functional failure [25]

1.3 Motivation

In micrometer-range technologies, interconnect was a minor influence on signal propagation but as the technology is shrinking into the nanometer domain, it is estimated that interconnect delays will account for 80% of the cycle time in digital circuits while the switching (set up time + hold time) of the gates will account for only 20% [3].

Increasing line resistance is the main reason behind the increasing interconnect delay. Resistance is inversely proportional to the cross-sectional area of the wire. Due to the rising need for higher densities on-chip, wiring pitches are dropping rapidly at about the same rate as gate length. In an effort to keep resistance from increasing too quickly, many processes are scaling line thickness (or height) at a slower rate, which results in taller, thinner wires.

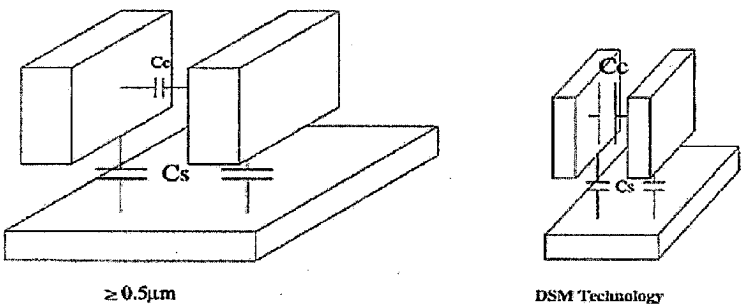


Figure 4 Metal wire aspect ratio change over technologies [25]

These high-aspect ratio lines have a detrimental side effect as they result in a large amount of coupling capacitance. With aspect ratio greater than one, lines tend to have more capacitance to neighboring wires than to upper and lower wiring layers (due to orthogonal routing between adjacent layers). In addition, spacing between wires is shrinking quickly in an attempt to maintain high packing densities, further increasing coupling capacitance.

Crosstalk is highly sensitive to the ratio of coupling capacitance to total capacitance, implying that it will become a larger issue as interconnect dimensions continue to scales. However, crosstalk at the global level will not be as significant as the local level due to the relatively large spacings and use of large repeaters—their capacitance will dampen the effects of coupling capacitance. So to deal with the crosstalk induced failures in the local interconnects, the modelling of crosstalk noise becomes an important issue.

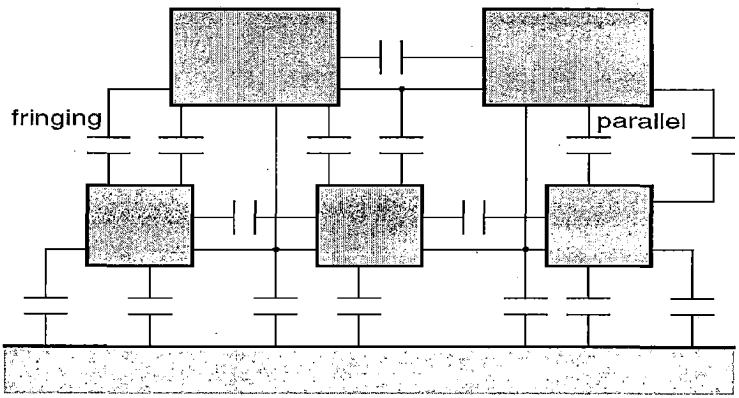


Figure 5 Capacitances between metal 1 and 2 lines [8]

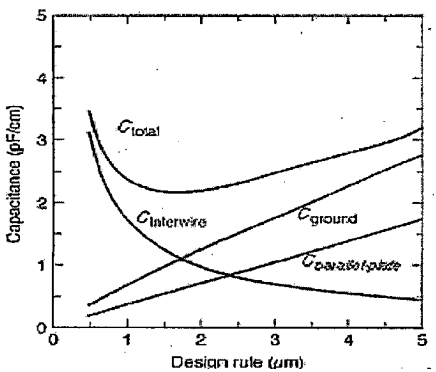


Figure 6 Capacitance variation with technology [8]

Most gate-level timing simulators are presently unable to accurately simulate "noise glitches" over a large number of circuit elements. Transistor-level circuit simulators such as SPICE are very computationally intensive and consequently require a significant amount of time to model noise on a small number of circuit elements. This makes it impractical to simulate the effects of noise on larger circuits. Hence what is needed is a methodology and a practical implementation technique for accurately and efficiently modeling the effects of electrical noise on larger circuits.

1.4 Hierarchical approach to crosstalk noise analysis

In a block-based hierarchical design style, the microarchitecture of the circuit can be decomposed into blocks of 50K--100K gates. The largest ASICs manufactured today are entire systems-on-a-chip. These SoCs typically consist of a number of intellectual-property (IP) blocks connected by means of standard on-chip buses. In the signoff phase of this hierarchical design flow, crosstalk noise analysis is usually undertaken to assure the designers of the integrated circuit that the circuit does not exhibit undue sensitivity to noise, or that the integrated circuit will not experience failures due to noise related problems.

Conventional hierarchical noise analysis of VLSI circuits is usually based on a two step method, which may consist of 1) transistor level noise analysis creating a noise abstract of a logic block and 2) chip level noise analysis using the noise abstracts created in step 1. Utilizing this technique, noise failures inside these logic blocks can only be detected at transistor level noise analysis, while noise failures at the chip level can only be found at chip level noise analysis.

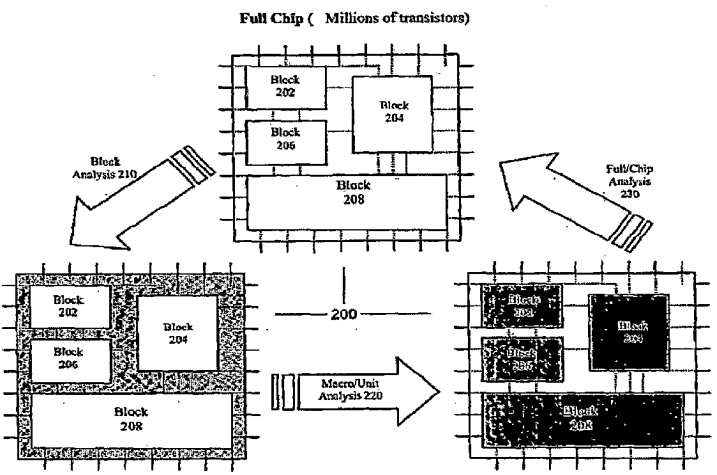


Figure 7 Hierarchical noise analysis

1.5 Major Contributions of this work

The key contributions of the dissertation are:

1. It proposes a hybrid model to represent coupling glitch which can take into account process variations.
2. It proposes a new model to account for the presence of propagated noise while calculating the peak of the glitch that a gate can withstand without causing a functional failure. The model thus includes the propagated noise in Noise Rejection Curve characterization. It also validates the linear superposition approach while adding the injected and propagated noise.
3. It proposes that the resultant effect of variation of independent parameters (process or circuit) on the delay noise can be approximated by taking the average of delays due to each parameter variation independently.

1.6 Dissertation Organization

Chapter 2 deals with noise characterization methodology, noise rejection curves and noise propagation characteristics.

Chapter 3 deals with different noise models and their drawbacks and proposes the new hybrid model.

Chapter 4 deals with statistical crosstalk analysis using the proposed hybrid model.

Chapter 5 deals with different models used for combination of propagated noise and injected noise and proposes a new model for calculation of peak noise.

Chapter 6 deals with delay noise fundamentals and results of deterministic and statistical analysis of delay noise.

Chapter 7 concludes the dissertation.

2. Noise characterization

2.1 Noise failure criteria

In the nanometer era, where chip level analysis tools require more data in ASIC libraries, ASIC cell library characterization time has increased drastically. More complex SPICE models, larger SPICE netlists due to 3-D and more accurate parasitic extraction, and a large number of PTV (Process, Temperature, and Voltage) conditions further contribute to increased characterization runtime. It is found that characterization of noise parameters alone takes more than 60% of the total library characterization time.

The switching of an aggressor signal net, which has capacitive coupling to a quiet victim net can cause functional failure at a victim receiver cell. However, the failure will occur only if the noise height (peak of the glitch) is higher than a threshold value, which depends on the width of the glitch.

In order to determine whether or not a certain input glitch to a gate can cause a failure, a simple DC noise margin is not sufficient. Noise immunity curve must be characterized to capture gate response to varying glitch width. For any input noise glitch width (glitch pulse duration) to the gate, noise immunity curve shows the minimum input glitch height (glitch peak voltage) required to cause a gate to fail.

To characterize a noise immunity curve, various noise waveforms with different widths and heights can be applied to the input of each cell and identify the smallest height for a given width that causes the cell to fail [9][10]. A Plot of these input glitch height-width pairs represents the noise immunity curve, which resembles a hyperbola especially if the width is derived from the area and height of the glitch. Any point above the noise immunity curve represents the height and width of an input glitch signal, which can cause the gate to fail.

2.2 Noise Parameters

The types of library data required are: (1) Steady State Driver Models (I-V Curves), (2) Noise Immunity Curves, and (3) Noise Propagation Characteristics. The need for these parameters is acknowledged by the representation in popular industry standards. Since

steady state I-V curves involve only DC simulations, the effort to create them is trivial in comparison to the other parameters.

2.2.1 Noise Immunity Curves (NIC)

The Noise Immunity Curve (NIC) is a representation of the amount of noise peak and width required at an input pin of a gate/cell, to cause a failure. There may be several ways to define the failure criteria, depending on the chip-level analysis methodology adopted. An NIC may be produced by applying a triangular voltage stimulus with various heights and widths to an input of the cell and observing the output voltage. If the output noise peak exceeds a predefined threshold value, the corresponding input peak is called the failure point, for that input noise width. The collection of such points for various noise widths constitutes the NIC. The NIC also depends on the output load and input states (or vectors). The input vectors affect the noise immunity because they select different signal paths to the output. Fig.8 shows the NIC for two different vectors of a combinational arc.

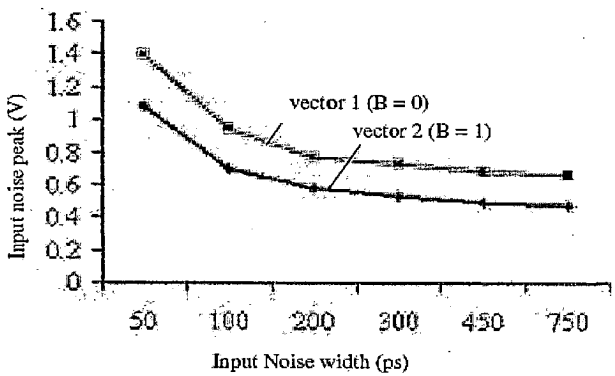


Figure 8 NIC of half adder for different vectors [9]

The NIC for sequential cells (flip-flops) is measured by testing for a change of state at the output in the presence of noise at the inputs. It involves a two-dimensional search: finding the minimum noise peak that causes a failure, and sweeping the clock position to meet the setup and hold constraints.

2.2.2 Noise Propagation Characteristics (NPC)

Noise propagation (NP) data represents the output glitch height and width for varying input glitch width and height. It is measured for every timing path/arc of combinational cells, by injecting an input glitch of known width and height, and measuring the corresponding output glitch height and width. This data is typically generated for several input glitch heights and input glitch widths, for each timing condition, and at various output loads.

2.3 Noise characterization methodology

Several transistor-level tools with various speed-up methods have been developed. However, all the known tools work either at transistor-level, or at gate-level but with strong dependency on SPICE. The approach is to characterize all significant noise properties of the digital circuit at transistor level and capture the data in a gate-level library. This allows full integration of the noise analysis with any existing static timing analysis tool.

First a set of input noise signals generated by a capacitively coupled circuit is obtained (Figure 9). Then, these generated signals are used to characterize each library cell that is used in the design for modeling noise immunity as well as noise propagation. The library cell under characterization is placed at the load of the victim net. The driver of the victim net is set to steady state (either logic zero or logic one). Then, changing various circuit parameters would cause various input noise glitches to be generated. These parameters include the input slew of the aggressor net drivers, ground and coupling capacitances and also the resistance of the nets. The drive strength (sizes) of the driver cells can also impact the noise signals.

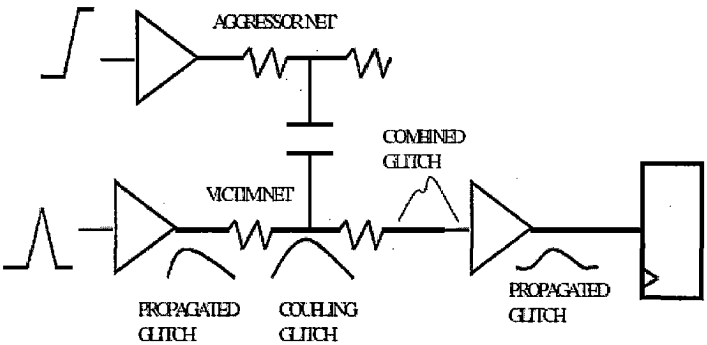


Figure 9 Illustration of the modelling of noise [24]

Another required characterization is the propagation of the noise through each cell in the library. This can be obtained by using the same circuit used for NIC characterization, and represent the propagated values as a lookup table indexed by input noise width, input noise height, output load, and also optionally time to peak of the input noise. Once the noise characterization is performed, the noise analysis consists of calculating the noise glitch induced by the switching of the aggressors and also using noise propagation tables to determine the amount of noise that will propagate and combine with the noise induced on the downstream logic.

Using the coupled circuit as the stimulus during the characterization is not practical because of speed and a large number of circuits are needed to cover most possible glitch shapes. To simplify this, triangular shaped waveforms can be used as stimulus instead. However, experiments show that such waveforms could have significant accuracy problems, and in many cases underestimate the propagated noise or the failure height used in NIC.

One way to make the glitch shape closer to a real circuit is to filter the triangular glitch using an RC circuit as illustrated in Figure 10. The RC circuit changes the height and width of the triangular waveform. Therefore, to accurately control height and shape of the smoothed glitch, a reverse transform for the RC filter must be obtained.

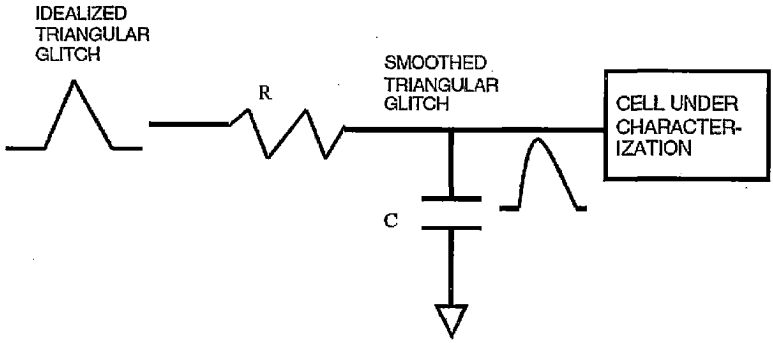


Figure 10 Cell characterization methodology [24]

The major disadvantage of the RC smoothing of triangular glitches is that the RC constant is dependent on the shape of the glitch. If the RC is too large then narrow glitch is filtered. If too small then glitch is not smooth enough. Furthermore, the RC filter can interfere with detailed parasitics inside the cell netlist thus making the reverse transform virtually impossible for some cells. These limitations and the complications due to varying RC for each glitch make this method impractical. Also, the RC-smoothed waveform does not have a shape that is close to that of real coupling glitches.

There are many ways to approximate the shape of real glitches. The key for accuracy is that the shape resembles real circuit glitches (not possible with triangle or trapezoid waveforms). The key for characterization is controllability – how to generate sufficient range of shapes and sizes (very difficult with real coupled circuits).

3. Noise modelling

3.1 Background work

With the increasing dominance of coupling capacitances in nanometer libraries and the growing complexity of system-on-chip designs, accurate yet fast crosstalk [2][3] noise analysis and avoidance methods have become necessary. The key to the effectiveness of such solutions lies in comprehensive modeling. To enable accurate analysis and identification of violations, noise parameters like noise immunity and noise propagation behavior are required to be characterized for cells in the library.

In [11], various failure criteria have been described. Regardless of the criterion, the process of generating the noise immunity remains the same. For example, one failure criterion is static gain failure criterion, which defines the failure as the point where the height of the propagated glitch through the receiver cell reaches the unity gain point of its DC transfer curve. However, to get all different types of input noise waveform shapes, an ideal triangle is commonly used instead of the real coupling waveform. This is because a real coupling network contains too many different parameters such as amount of coupling, ground capacitance, or driver sizes, which are not easily controllable to allow accurate characterization of noise immunity curves.

Using triangular or trapezoid approximation, instead of real waveforms can underestimate the propagated height by as much as 20% and as a result, the analysis tools could miss a real violation [12]. Also, other noise waveform models, such as 2-pole waveforms can significantly overestimate the propagated height, which can also be problematic as it causes the analysis tools to report or attempt to fix too many false failures.

To avoid these inaccuracies, the use of a more pessimistic input noise waveform such as an isosceles triangle has also been suggested, but even that could not solve the problem of underestimated propagated noise height.

The propagated noise height from both the approximated triangle and the isosceles triangle underestimate the real waveform. This is mainly because the triangle has less area near the peak than a real noise waveform induced by capacitive coupling, and this portion contains the effective energy that produce output pulse.

In [13] the authors have proposed two techniques of using n-pole equation where n is set to 2 and a trapezoid waveform approach to model the crosstalk noise. However, the n-pole model is complicated and is not computationally efficient, therefore it is generally impractical.

3.2 Two pole model

For a simple two pole model as shown by following equation:

$$f(t) = a1.e^{p1.t} + a2.e^{p2.t}$$

The 4 coefficients, al, a2, p1, and p2, can be available by solving the following equations for given glitch height (H), width (W, based on area), and time-to-peak (t_0).

$$f(0) = 0$$

$$f(t_0) = H$$

$$f'(t_0) = 0$$

$$\int_0^{\infty} f(t) = W.H/2, \qquad \int_0^{t_0} f(t) = t_0.H/2$$

where t_0 is the time the waveform reaches its peak.

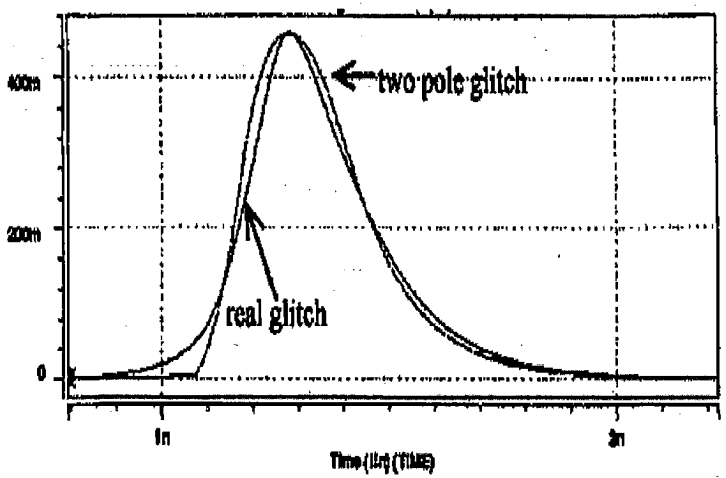


Figure 11 2-pole glitch vs real glitch [12]

However the waveform produced by two pole equation does not resemble the real coupling glitch well as it is shown in figure 11 and may over estimate the propagations. Also, since it is not easy to derive each of the 4 coefficients to be functions of basic glitch characteristics, the flexibility of fine tuning glitches which is required in noise immunity characterization is restricted. Increasing the number of poles to improve the quality of the waveform is not a good option either, as adding more parameters will significantly increase the difficulty of deriving parameters for this model. So this method is generally not preferred in practical.

3.3 Triangular model

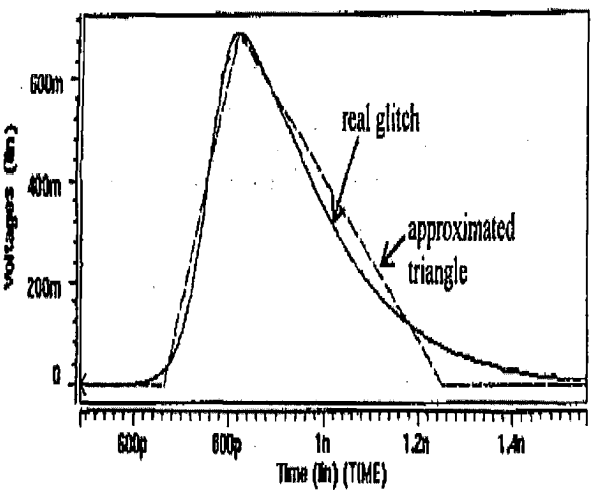


Figure 12 triangular glitch vs real glitch [12]

The propagated noise height from the approximated triangle underestimates the real waveform. This is mainly because the triangle has less area near the peak than a real noise waveform induced by capacitive coupling, and this portion contains the effective energy that produce output pulse.

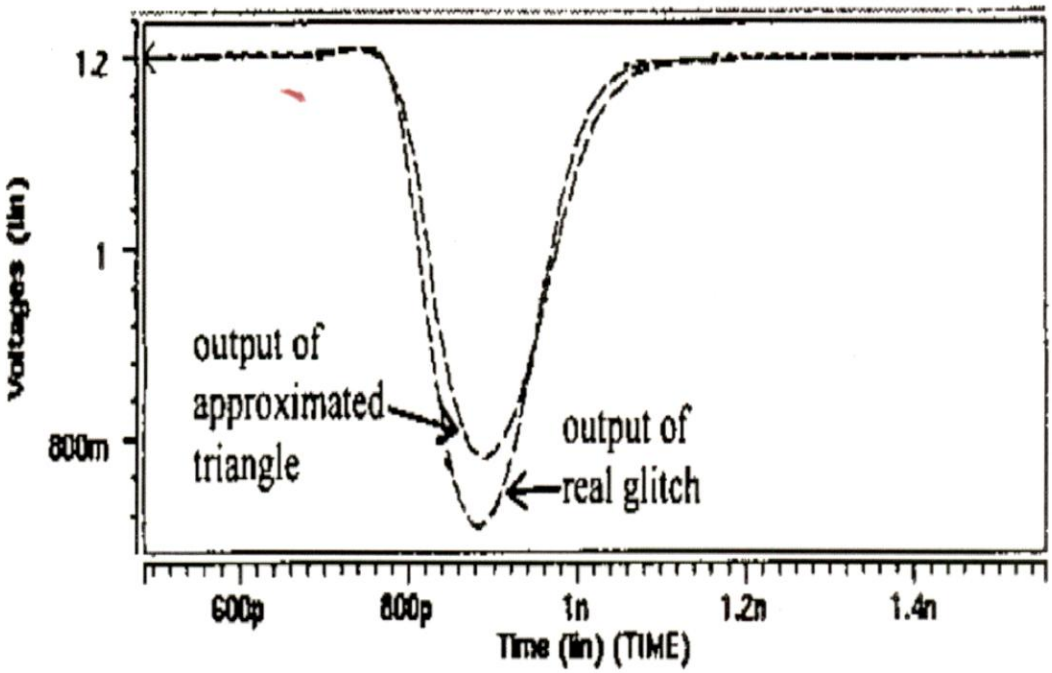


Figure 13 Propagated triangular glitch vs real glitch [12]

3.4 Trapezoidal model

The other method from [12] is piecewise linear trapezoid model. As shown in Figure 14, the trapezoid model determines its height from the input waveform when the output waveform reaches its peak.

This model overestimates the waveform area around the leading edge of the glitch, but also underestimates the important area near the peak of the glitch. Obviously, the overestimation and underestimation do not have a balanced effect and can cause significant underestimation, as it is shown in Figure 14.

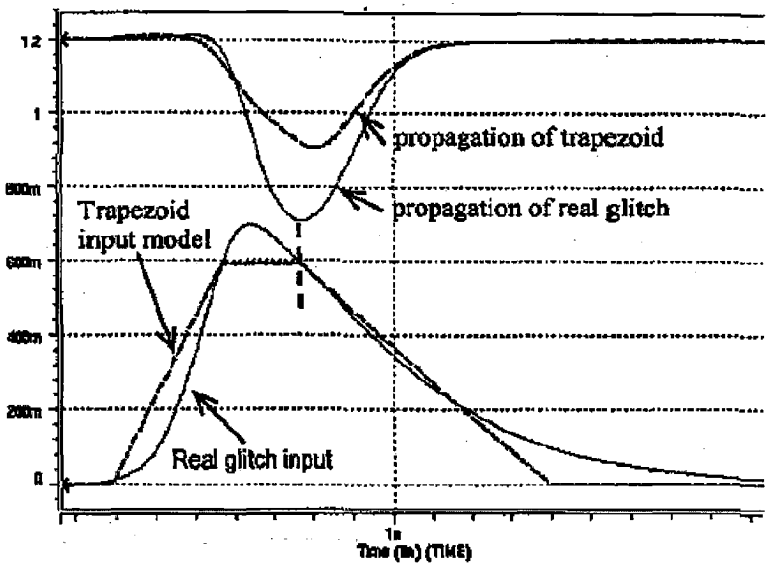


Figure 14 Trapezoidal glitch vs real glitch [12]

3.5 Weibull model

In [12] the authors try to fit a four-parameter Weibull model with the basic characteristics of a coupling-induced noise glitch, and such model can have very good accuracy compared to real waveforms. The basic characteristics of a coupled glitch are shown in Figure 15. The height of the glitch (H) is the peak voltage of the waveform. The area of the glitch (A) is the area underneath of the glitch waveform. Another important characteristic is the ratio of the area up to the peak of the waveform divided by the total area of the waveform, which we refer to as peak-area-ratio (P).

$$P = \frac{A_p}{A}$$

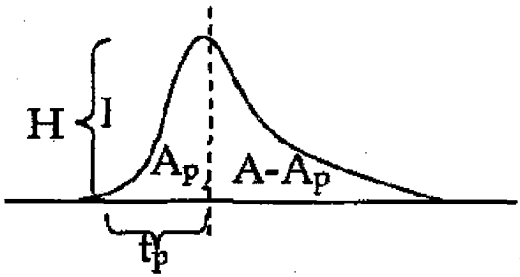


Figure 15 Characteristics of a coupling noise waveform [12]

There are many different criteria for defining the width of the glitch (W) such as the 5% crossing of the peak voltage; however the tail end of the noise waveform can be very long in certain cases, which makes it impossible to measure the width correctly.

To avoid this inaccuracy, the width of the waveform is defined as twice the area divided by height. This method of measuring width has no ambiguity as both the area and height of the glitch can be accurately measured in SPICE and is also consistent with the historical definition of the width for triangular waveforms.

The Weibull function has been widely used as a lifetime probability distribution in reliability engineering. The Weibull Probability Density Function (PDF) can look similar to following: the noise glitch for a certain range of parameters. The Weibull Cumulative Density Function (CDF) resembles a regular gate transition and can be used to describe waveforms generated by uncoupled RC networks.

In [12], the four-parameter Weibull PDF is used to match the crosstalk glitch waveforms induced by aggressor net to a quiet victim. The four-parameter Weibull PDF is defined in the following manner:

$$c > 1, a > 0, b > 0, t \geq 0$$

$$f(t) = a.\left(\frac{c-1}{c}\right)^{\frac{1-c}{c}}.\left(\frac{t-t_0}{b}+\left(\frac{c-1}{c}\right)^{\frac{1}{c}}\right)^{c-1}.e^{-\left(\left(\frac{t-t_0}{b}+\left(\frac{c-1}{c}\right)^{\frac{1}{c}}\right)^{c}-\frac{c-1}{c}\right)}$$

The four parameters, a, b, c, $t0$, are defined as follows.

a is the glitch height

$t0$ is the peak time of the glitch (time-to-peak of the glitch when glitch starts at 0 and can be used to shift the signal in time)

c is a shape coefficient which can be derived from the waveform's peak-area ratio (P).

$$c = \frac{1}{1 + ln(1 - P)}$$

b is a glitch width coefficient,

$$b = glitch_{width} \cdot \frac{c}{2} \cdot \left(\frac{c-1}{c}\right)^{\frac{c-1}{c}} \cdot e^{\left(\frac{1-c}{c}\right)}$$

The propagated glitch due to the coupling glitch and the Weibull glitch are found to be practically overlapping.

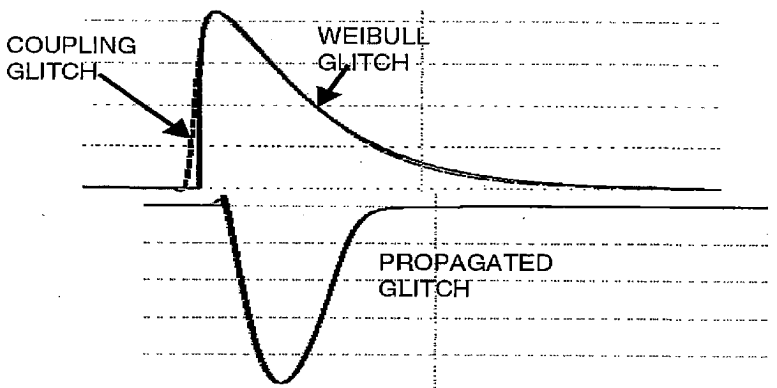


Figure 16 Weibull glitch vs real glitch [24]

Limitations of the Weibull model

1- The optimization of the shape coefficient required makes it suitable only for library characterization and not for real time simulations.
2- It has a higher complexity compared to triangular or 2-pole model.
3- The model cannot take into account process variations.

3.6 Observations

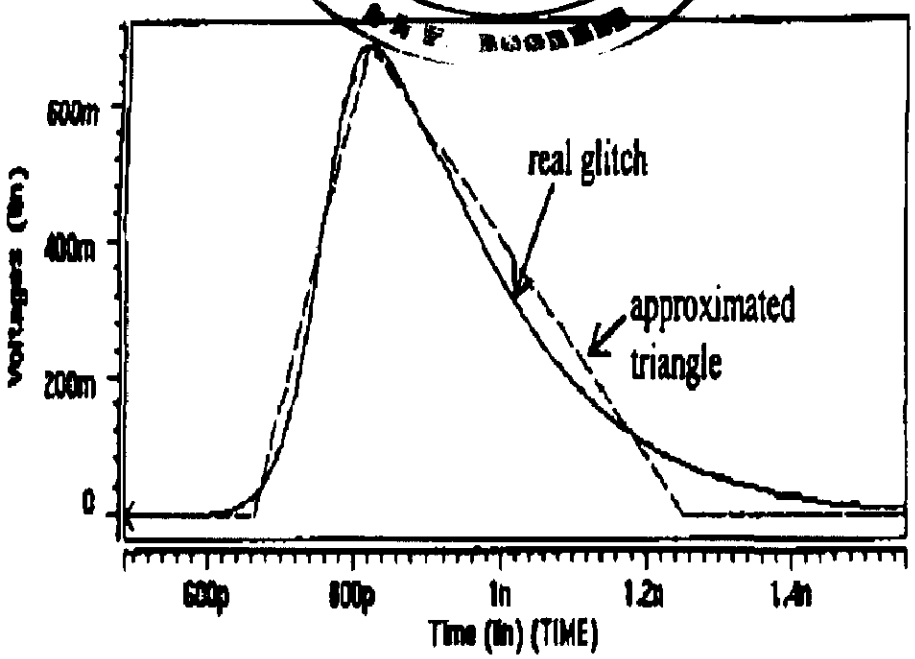


Figure 17 Triangular glitch vs real glitch [12]

Rising half of a real coupling glitch is similar to that of a triangular glitch

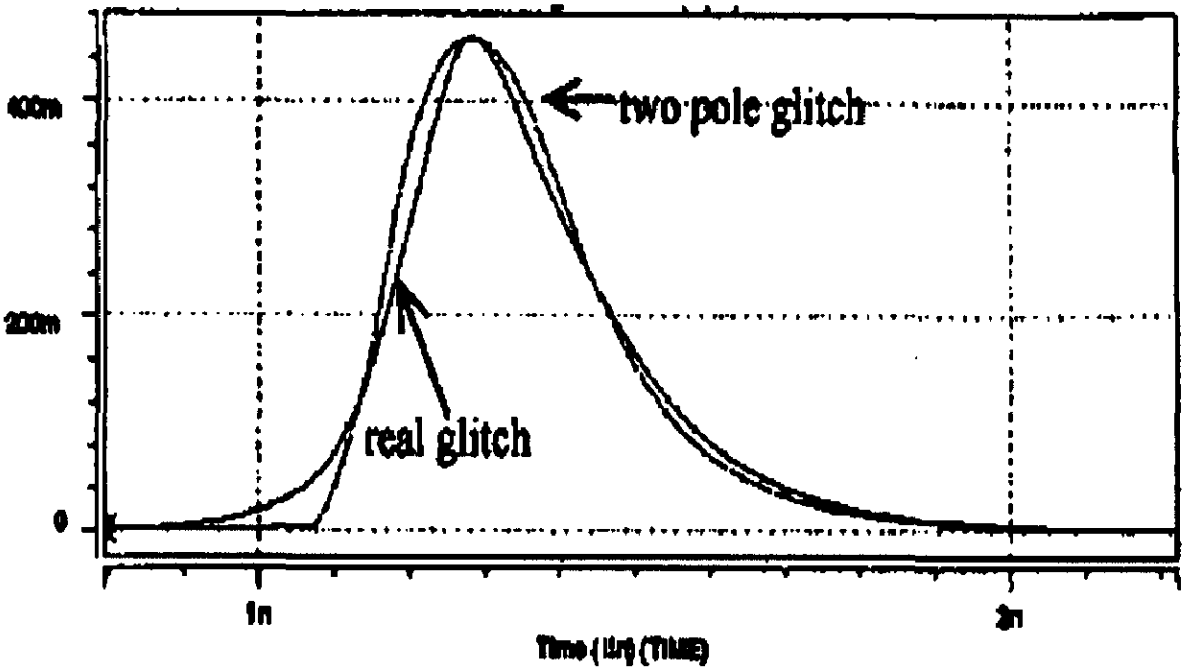


Figure 18 2-pole glitch vs real glitch [12]

Falling half of a real coupling glitch is similar to that of the 2-pole glitch.

Using these observations we propose a hybrid model to represent the coupling glitch.

3.7 Proposed Hybrid model

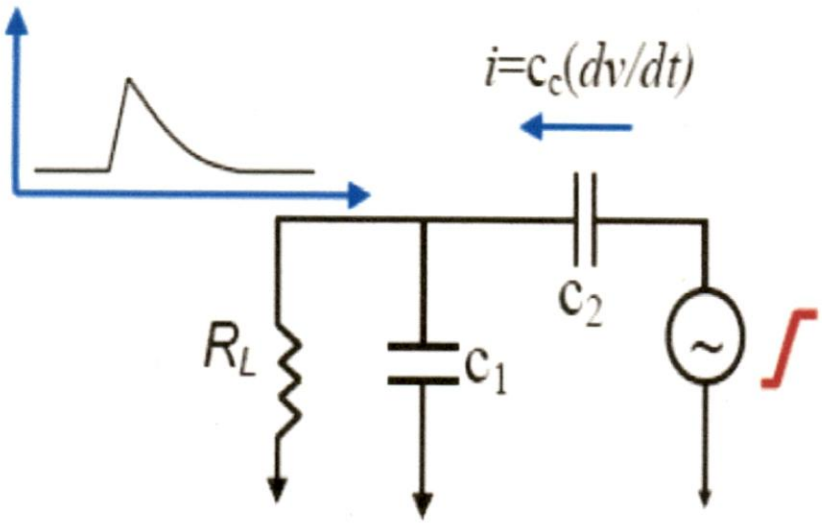


Figure 19 Equivalent model of aggressor- victim pair [14]

The equivalent model [14] shows the load capacitance as C_1, coupling capacitance as C_2 and the load resistance as R_L . The proposed model of coupling glitch is shown in Fig 20.

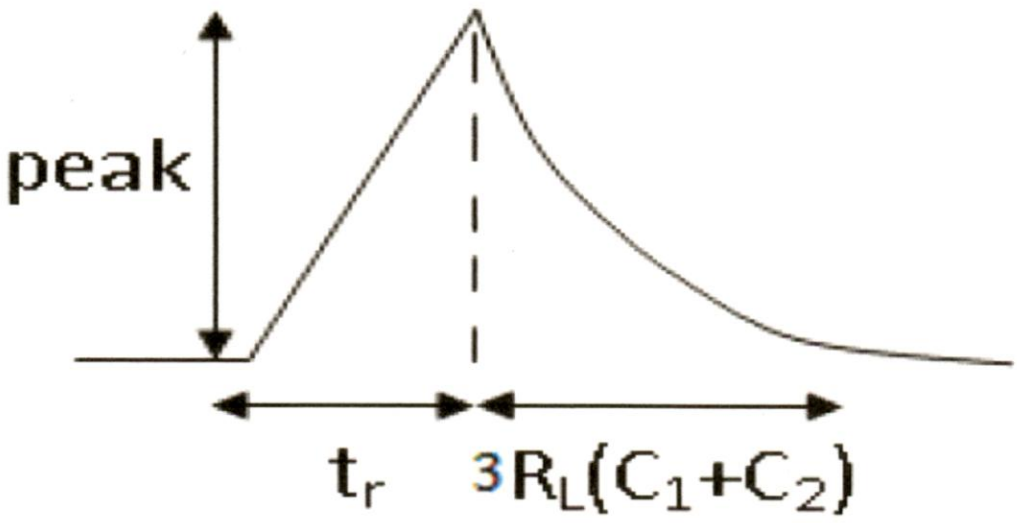


Figure 20 Hybrid model of glitch (proposed)

The width of the glitch is defined as

$$W = t_r + 3R_L(C_1 + C_2)$$

The discharging time is taken to be approximately 3 time constants.

The rising part of the glitch is similar to that of the triangular waveform and the falling part is similar to that of the 2-pole waveform.

3.7.1 Advantages of hybrid model

1. It is simpler than Weibull model.
2. It can take into account process variations i.e. Principal Component Analysis parameters.
3. It builds upon sensitivities given by SSTA library characterization.
4. It is best suited for statistical crosstalk analysis.

3.7.2 Disadvantages of hybrid model

1. It is less accurate than Weibull model.
2. It can be used for lumped RC models only.

4. Functional noise analysis

4.1 Introduction

With dramatic increase of crosstalk noise in modern deep-submicron designs, we require more accurate noise estimation and noise failure analysis. Over-pessimistic analysis results in a number of false noise failures that require excessive chip resources to be fixed or avoided. On the other hand, with scaling process nodes the ability to control critical device parameters is becoming increasingly difficult and process variations have to be taken into account for all sub 90nm designs.

A lot of work has been done in the field of Statistical Static Timing Analysis by factoring in the effect of process variations in static timing analysis [15]. One of the primary results that the user gets from using the technique of SSTA is the timing yield of the design, in terms of percentage of functional chips at a given frequency. Static crosstalk analysis is inherently more complex in nature than STA. In the UDSM era, it is imperative to consider the effects of process variations in crosstalk and static noise analysis to get a better functional yield of the design.

Aggressor slew is a critical parameter in crosstalk analysis. In [14] a method was proposed for incorporating the effect of process variations into the aggressor net slew, and on the subsequent impact on a key metric of crosstalk - the peak of the glitch caused in the victim net. Peak of glitch is undoubtedly an important metric of crosstalk noise but when it comes to propagation of noise through a logic gate and checking it for failure condition, the glitch width or the area under the glitch becomes a more important parameter.

The effect of process variations on the area of the glitch on the victim net is looked into and a method is proposed to use an SSTA like approach in statistical crosstalk analysis. With a minimum incremental effort of characterization, we can use the same statistical libraries to represent the Noise Rejection Curves as an indirect function of process parameter variations thus reducing the false noise failures and hence pessimism in crosstalk analysis.

4.2 Background work

[14] proposes a framework to address the effects of process variations on crosstalk analysis in a manner similar to that used in statistical timing analysis. It looks at how the impact of variations of the aggressor slews can be factored into crosstalk analysis. However it doesn't consider the effect of glitch width or area on the noise failures.

The simple model of [14] is used as a baseline to show that not only glitch peak but also area under glitch varies linearly with the slew. It is known from previous work in the area of SSTA that slews and load capacitances vary linearly with device parameter variations; hence with chain rule we can conclude that glitch peak and area vary linearly with device variations. Thus we can calculate the sensitivities of different logic gates to different process variations for propagated noise calculation and functional failures thus moving towards a statistical static noise analysis (SSNA).

Following the SSTA approach, these sensitivities can be propagated and PDF of the propagated noise can be constructed at the endpoints using the computed sensitivity at these points.

Statistical libraries of cells are obtained in a precharacterization step in SSTA which contain the sensitivities of delay, slew and load capacitances to various process parameters. By adding the sensitivities of glitch peak and area to the same statistical libraries we can provide a statistical representation of the NRCs.

4.3 Statistical Functional Noise Analysis

In the Fig 21, we have a victim net held low by an inverter. The victim net is also coupled to another net called aggressor, with a rising waveform. The equivalent model shows the load capacitance as C_1, coupling capacitance as C_2 and the load resistance as R_L.

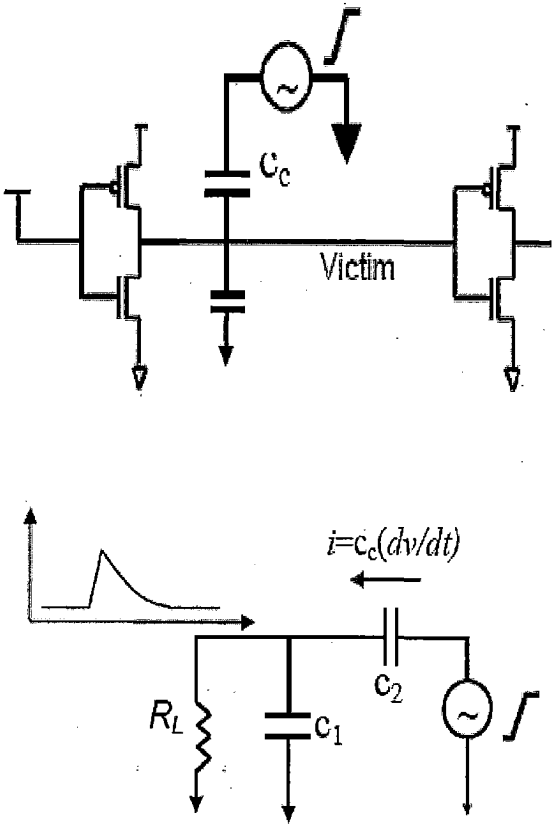


Figure 21 Coupling glitch calculation [14]

From [14], we have glitch peak in Fig. 21 as,

$$peak = \frac{V_{DD}R_LC_2}{t_r}\left(1 - exp\left(-\frac{t_r}{R_L(C_1 + C_2)}\right)\right)$$

where V_{DD} is the supply voltage and t_r is the slew of attacker net.

Let, due to process variation in parameter p, slew becomes $t_r+S\Delta p$, where S is the sensitivity of slew to change in process parameter. Hence new peak will be,

$$peak' = \frac{V_{DD}R_LC_2}{t_r + S\Delta p}\left(1 - exp\left(-\frac{t_r + S\Delta p}{R_L(C_1 + C_2)}\right)\right)$$

Hence, change in peak

$$\Delta peak = peak' - peak$$

From [14],

$$\Delta peak = \beta\Delta p \quad (1)$$

where β is the sensitivity of glitch peak to process variation.

From the proposed hybrid model, the glitch width can be divided into two parts, the rising time and the falling time. The rise time is equal to t_r and the fall time is equal to $3R_L(C_1 + C_2)$ where C_1 is the load capacitance and C_2 is the cross coupling capacitance.

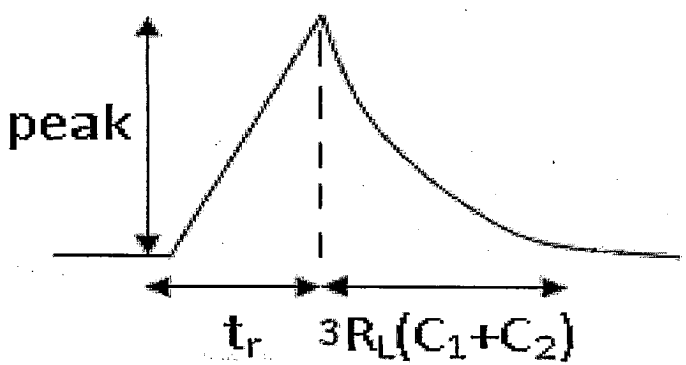


Figure 22 Components of glitch width from hybrid model

Now the width of the glitch can be written as

$$W = t_r + 3R_L(C_1 + C_2)$$

Now,

$$area = {}^1/_2 \times peak \times width$$

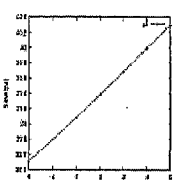
$$\Delta area = {}^1/_2 \times \Delta peak \times width + {}^1/_2 \times peak \times \Delta width$$

Substituting the value of Δpeak from (1)

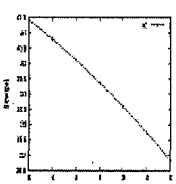
$$\Delta area = {}^1/_2 \times width \times \beta \times \Delta p + {}^1/_2 \times peak \times \Delta(t_r + 3R_L(C_1 + C_2))$$
$$= \beta' \times \Delta p + {}^1/_2 \times peak \times (\Delta t_r + \Delta 3R_L C_1 + \Delta 3R_L C_2)$$
$$= \beta' \times \Delta p + {}^1/_2 \times peak \times (\Delta t_r + 3R_L \Delta C_1)$$

(assuming change in R_L and C_2 with process variations (PCA parameters) to be negligible).

It is already known from [16], that slew varies linearly with change in process parameters. Fig. 23 shows variation in slew with change in different process parameters. A1-A4 are orthogonal parameters obtained after performing Principal Component Analysis (PCA) technique on process parameters.



(a) Param A1



(a) Param A2

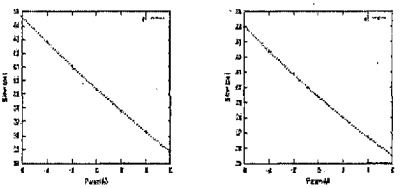


Figure 23 Slew changes with change in PCA process parameters (A1, A2, A3, A4) [16]

Hence,

$$\Delta t_r = k_1 \Delta p$$

Also from [16], load capacitance is found to vary linearly with change in process parameters.

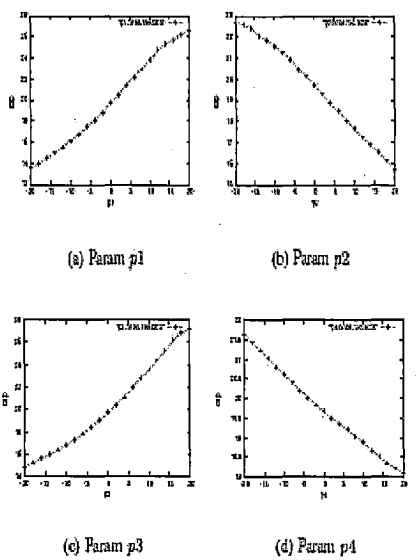


Figure 24 Load capacitance changes with change in PCA process parameters [16]

Hence,

$$3R_L\Delta C_1 = k_2\Delta p$$

Substituting the values of Δt_r and $5R_L\Delta C_1$,

$$\Delta area = \beta' \times \Delta p + {}^1/_2 \times peak \times (k_1\Delta p + k_2\Delta p)$$
$$= K\Delta p$$

where

$$K = \beta' + {}^1/_2 \times peak \times (k_1 + k_2)$$

So we can conclude that, under the given assumptions, area under a glitch varies linearly with process variations.

4.4 Results of Functional noise analysis

We did simulation on the basic prototype model and found that both glitch peak and glitch area varies linearly with slew. Glitch area also varies linearly with load capacitance.

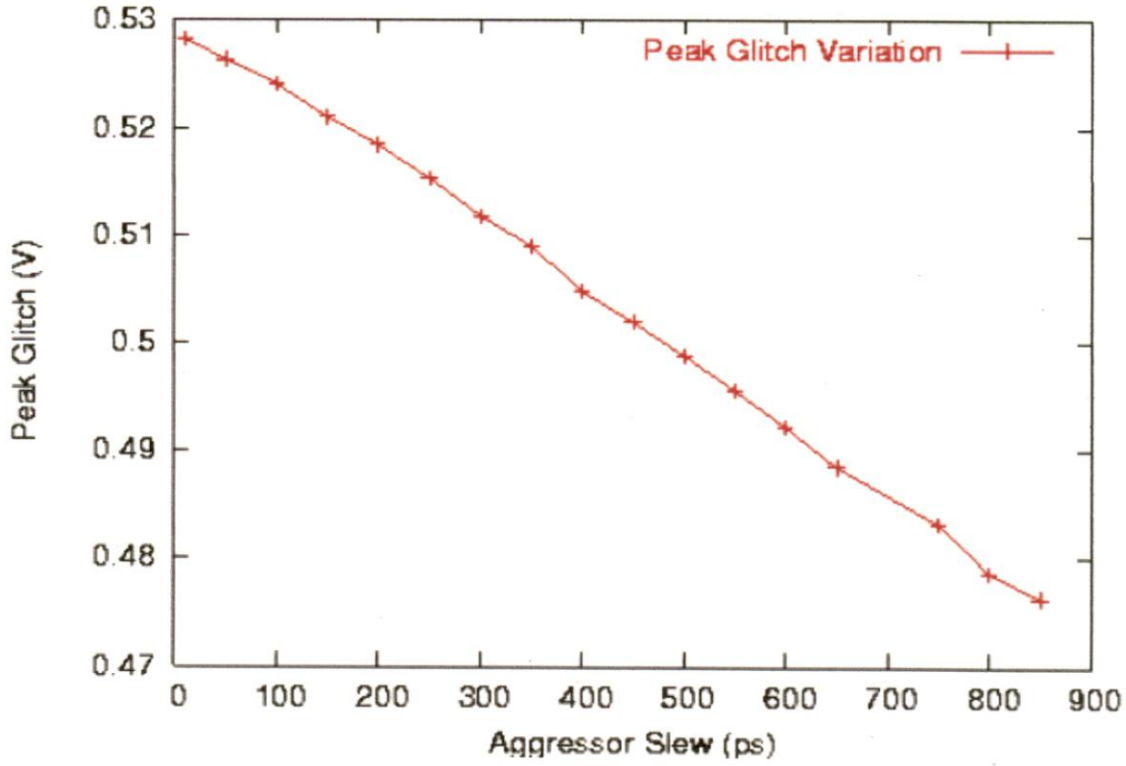


Figure 25 Glitch peak variation with change in slew [14]

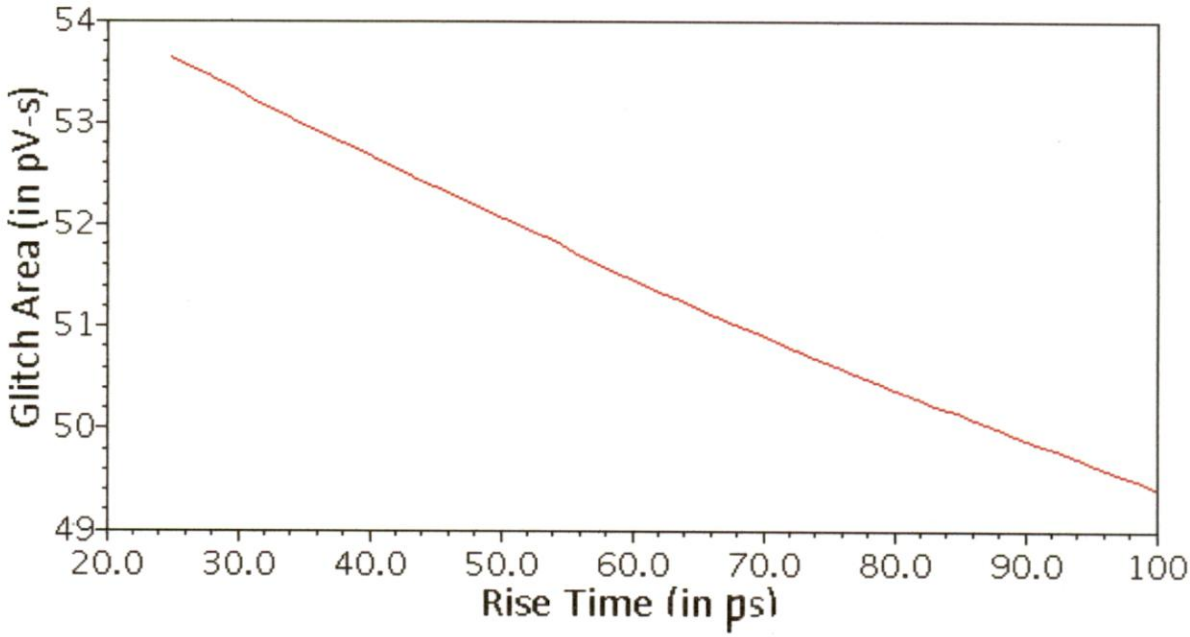


Figure 26 Glitch area variation with change in slew for load capacitance 20fF

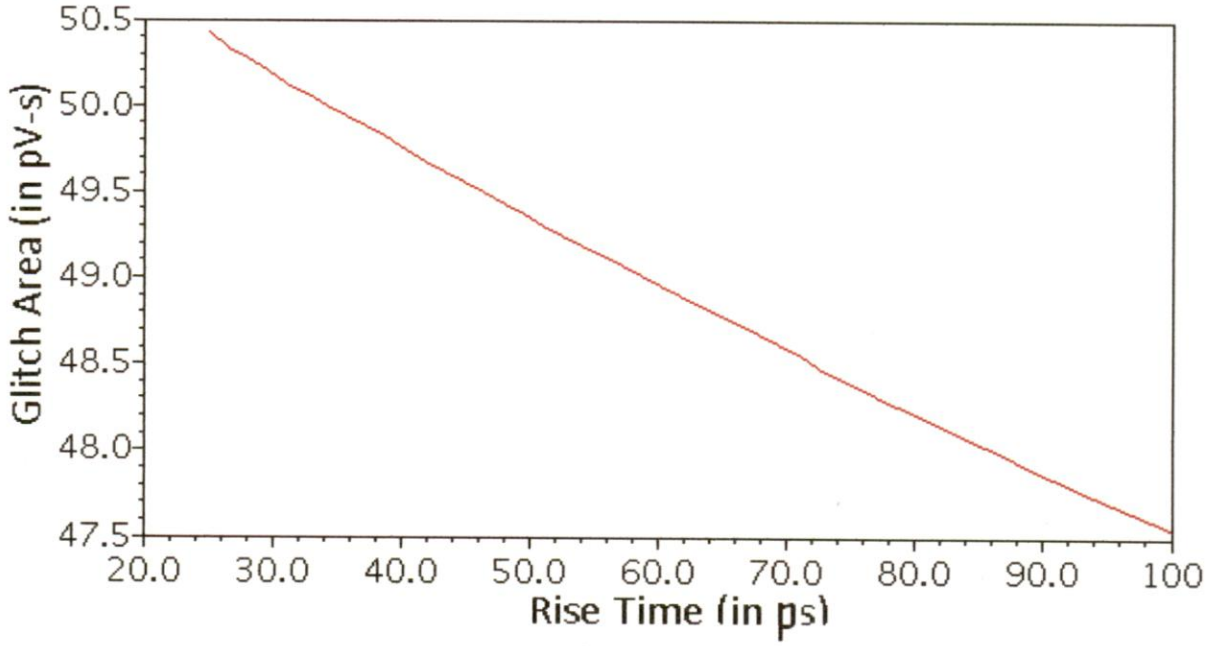


Figure 27 Glitch area variation with change in slew for load capacitance 30fF

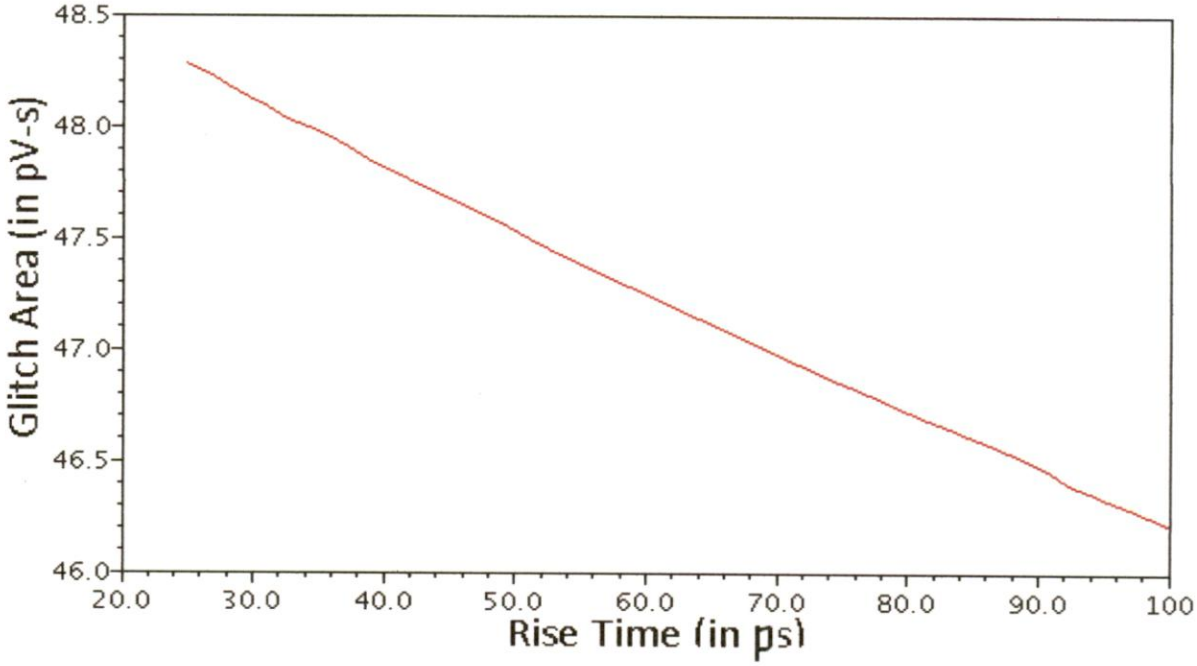


Figure 28 Glitch area variation with change in slew for load capacitance 40fF

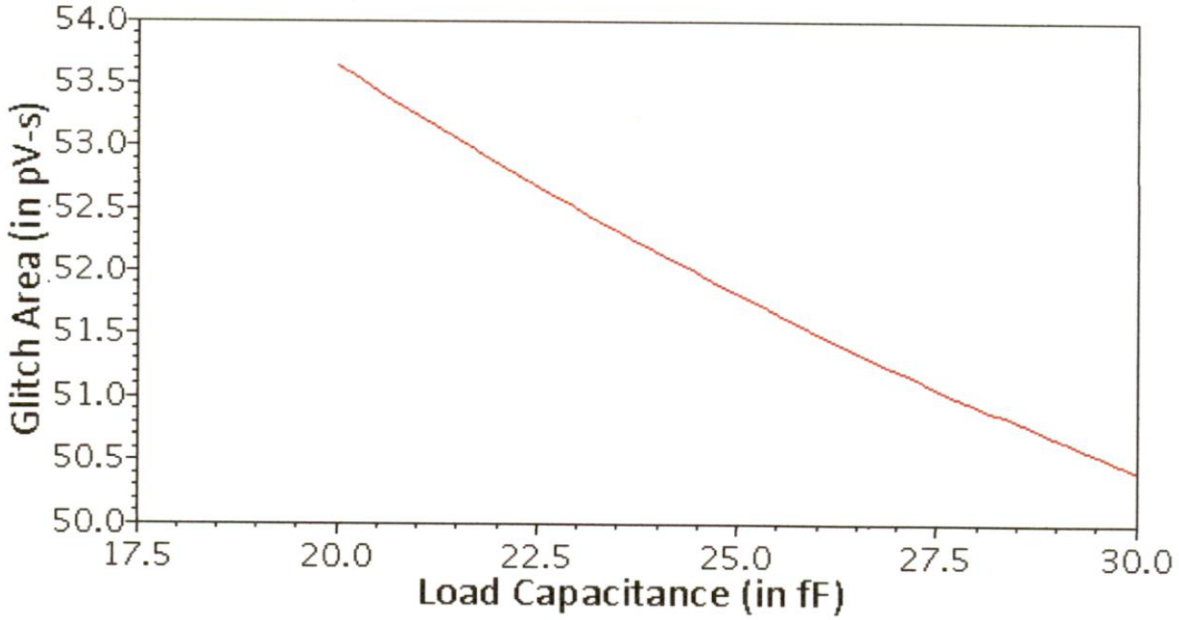


Figure 29 Glitch area variation with change in load capacitance for slew 25ps

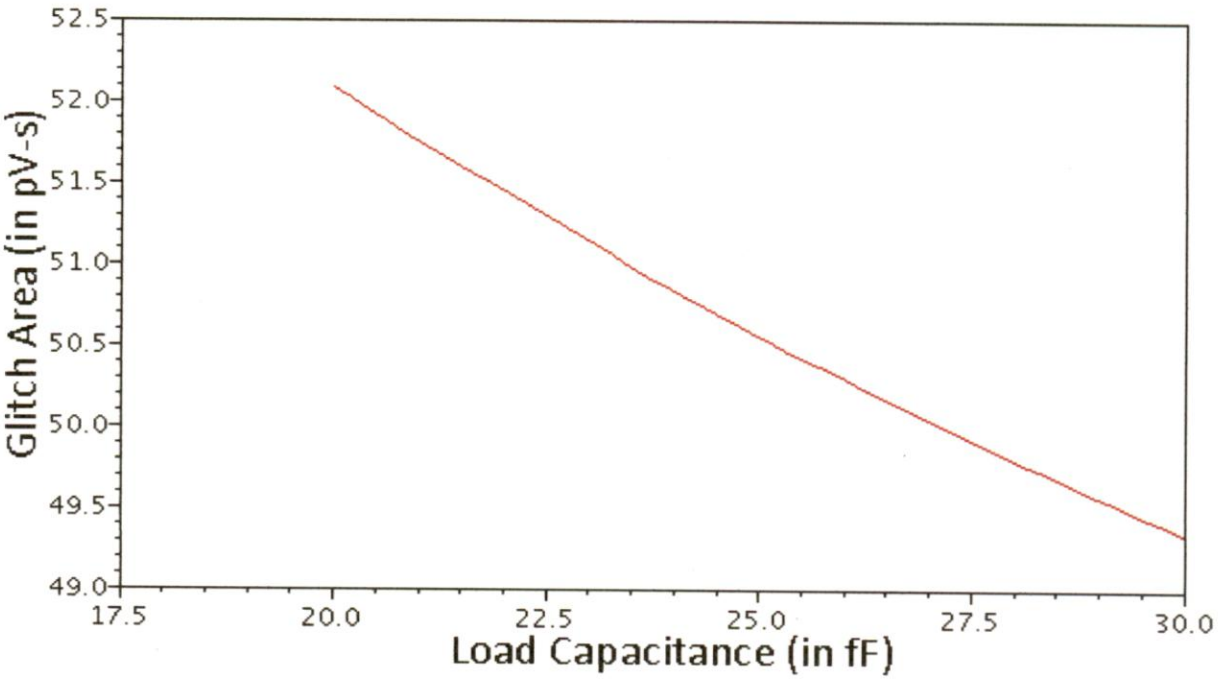


Figure 30 Glitch area variation with change in load capacitance for slew 50ps

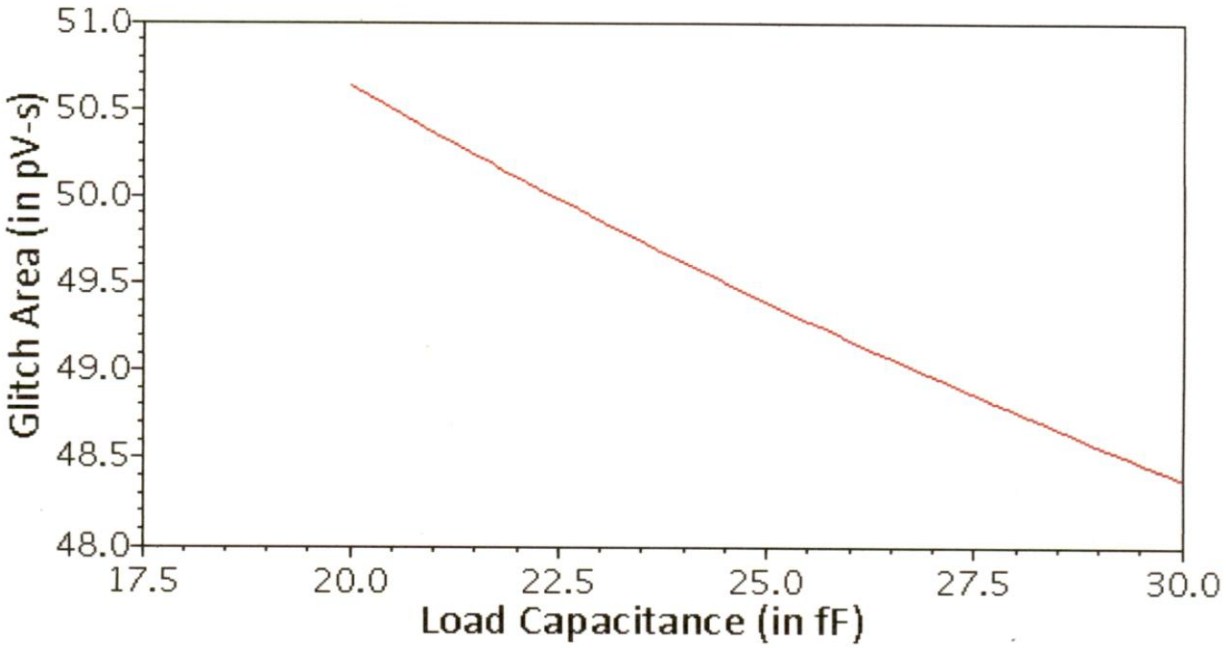


Figure 31 Glitch area variation with change in load capacitance for slew 75ps

Using chain rule [17], glitch peak can be represented as linear function of process parameter. Hence sensitivity of peak glitch to process parameter will be

$$S = \frac{\partial P}{\partial p} = \left(\frac{\partial t_r}{\partial p}\right)\left(\frac{\partial P}{\partial t_r}\right)$$

and sensitivity of glitch area to process parameter will be K which is the slope of the curve obtained.

Using both these sensitivities we can represent the peak and area of glitch as a probability distribution rather than the deterministic way in which noise rejection curves are plotted thus representing the noise rejection curve of any logic gate in a statistical manner. These sensitivities can be calculated in a pre-characterization step once and are not required to be computed again and again. This step can be integrated with characterization of delay, slew and load as functions of process parameters and can be included into statistical library. Thus using the same library we can go for SSTA as well as Statistical Static Crosstalk Analysis with minimum incremental effort required for noise characterization.

5. Propagated noise analysis

5.1 Introduction

Functional noise analysis tools must perform a conservative analysis to ensure that no possible noise problems remain undetected in the design. Noise injected by the aggressor nets combines with noise propagated from the input of the victim driver gate, as illustrated in Fig. 32. The combined noise pulse at the victim receiver is compared against noise rejection curve to determine if the particular noise pulse height and width results in a failure. In order to perform a conservative noise analysis, the noise peaks of the propagated noise and the injected noise are aligned to create a noise pulse with a maximum possible noise pulse height. In some noise analysis approaches, the propagated noise is treated as DC noise, which eliminates the problem of alignment, but results in a more pessimistic analysis.

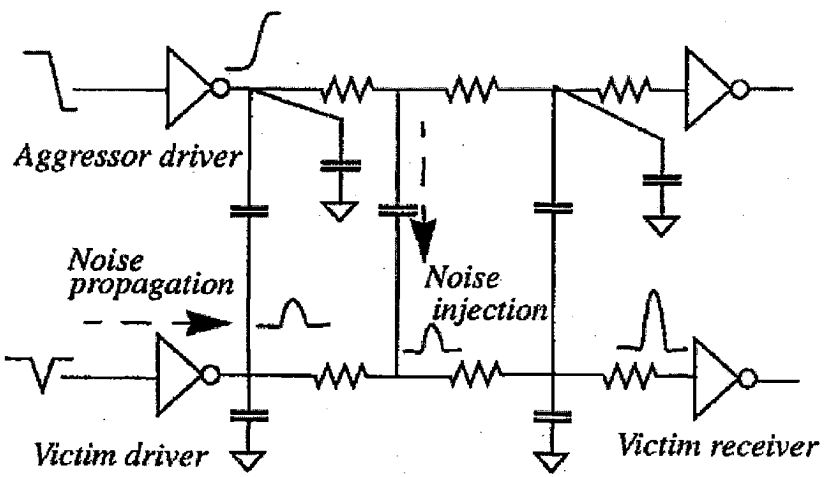


Figure 32 Noise due to propagation and injection [11]

5.2 Linear model

To efficiently compute the injected noise, analysis tools typically use linear models for the victim and aggressor driver gates, as shown in Fig. 33. The aggressor driver is represented with a Thevenin model, consisting of a ramp voltage source and Thevenin resistance R_A, providing the same signal slope as the original aggressor driver. The victim driver gate is modeled with a grounded resistance, called the holding resistance R_H.

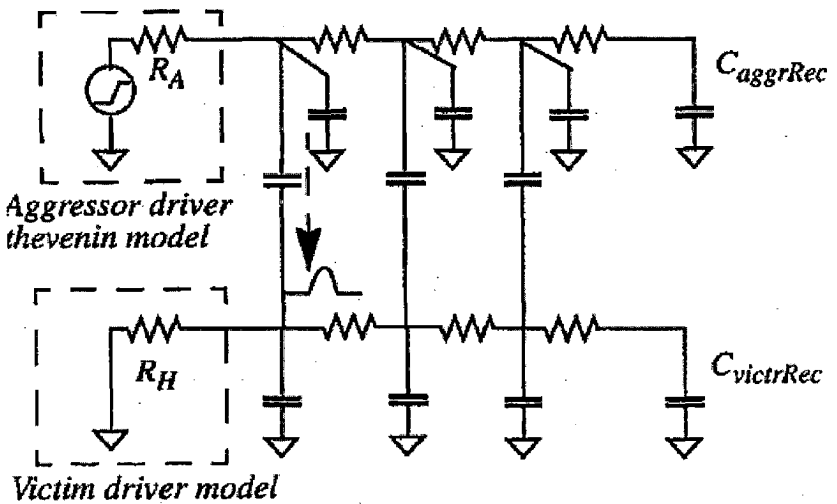


Figure 33 Linear model for injected noise [11]

This resistance is computed using a small signal analysis of the driver with both driver input and output biased at stable supply voltages, i.e., V_{DD} (GND) at the driver input and GND (V_{DD}) at the driver output). The use of such a linear model has several advantages. First, the entire circuit can be analyzed using efficient linear methods, such as reduced order modeling. Second, superposition can be used to sum the noise injected from each individual aggressor, making it simple to align the noise pulse peaks from each aggressor.

The propagated noise through the victim driver is typically computed using pre-characterized tables. The driver gate is simulated under a number of different noise pulse heights, pulse widths and loading conditions. For each condition, propagated noise is

computed using non-linear simulation and stored in a table. During noise analysis, the propagated noise is determined from this table based on the noise pulse height and width at the victim driver input and is added to the injected noise from aggressor nets. Since the injected and propagated noise pulses are added linearly, their worst-case alignment is again easily determined. In certain approaches, a pre-determined worst-case propagated noise is used for each gate, instead of the actual propagated noise. The worst-case propagated noise is defined as the maximum noise at the output of the driver in response to any input noise that does not cause a noise failure of the gate. This simplification improves the efficiency of the analysis while increasing its pessimism.

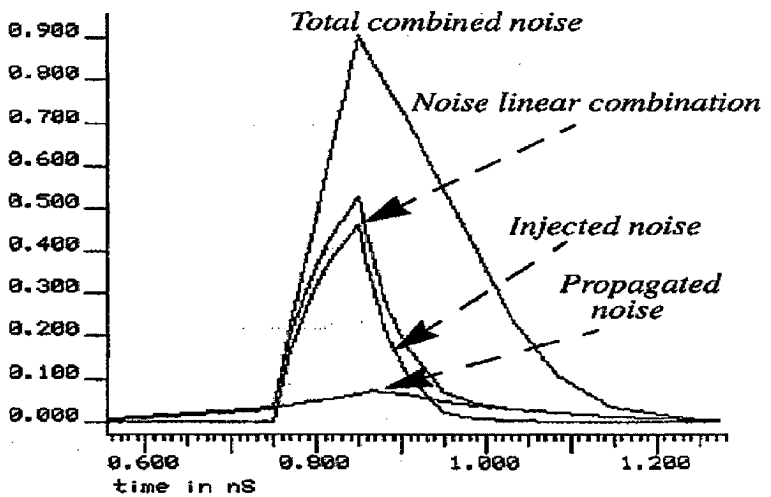


Figure 34 Comparison of combined propagated and injected noise with its linear combination [11]

Due to the use of linear addition of the injected and propagated noise pulses, propagated and injected noise can be computed independently, allowing for very efficient and simple analysis. This explains the wide spread use of this approach in commercial tools. It is based on an underlying assumption that the victim driver gate is linear and the approach is similar to noise analysis in analog circuits, where noise sources are small and devices exhibit relatively linear behavior. In digital circuit, on the other hand, noise can

be quite large (due to its inherent robustness), and the devices are constructed to have a very high gain and exhibit highly non-linear behavior. Therefore, the linear addition of propagated and injected noise is not valid and can result in a significant error in the computed noise.

Figure 34 shows the simulation results of a typical noise cluster from an industrial 0.13 micron design [11]. The propagated noise pulse has a height of 70 mV and the injected noise pulse a height of 453 mV. Therefore, the linear combination of the propagated and injected noise has a height of 523mV.However, non-linear simulation of the noise cluster results in a noise pulse with a height of 900 mV. This is due to the fact that the holding resistance of the victim driver is not constant during the noise propagation. Even though the propagated noise was small (70 mV), the holding resistance of the driver gate was significantly increased due to the noise at the driver input and output. In fact, it is possible that the input noise at the victim driver is sufficiently small such that it does not yield any propagated noise, while still significantly modulating the holding resistance and increasing the injected noise on the victim net.

It is clear that the linear combination of the propagated noise and injected noise used in existing analysis tools results in a significant underestimation of the actual noise. The straightforward approach to solving this problem is to perform non-linear simulation of the entire coupled interconnect and driver network. However, this approach has two serious difficulties. First, non-linear simulation is too slow for analysis of large design, even though the linear portion of the network can be represented with a reduced order model. Second, determining the worst-case alignment between the propagated noise and the injected noise is difficult in non-linear simulation, and typically involves expensive iterative search.

5.3 Victim driver Thevenin model

[11] proposes a new linear model, shown in Fig. 35 for accurate computation of the combined injected and propagated noise. In this model, the victim driver is represented with a Thevenin model consisting of a pulsed voltage source V_{ThPr} and resistance R_H.

These model parameters depend on the victim driver input noise v_{in}, as well as the total combined output noise v_o, in order to capture non-linearity of the victim driver.

The proposed methods use the DC-operating characteristics of the driver gate which is easily pre-computed and stored in a compact table. The approach therefore lends itself well for use in a pre-characterized cell based design flow.

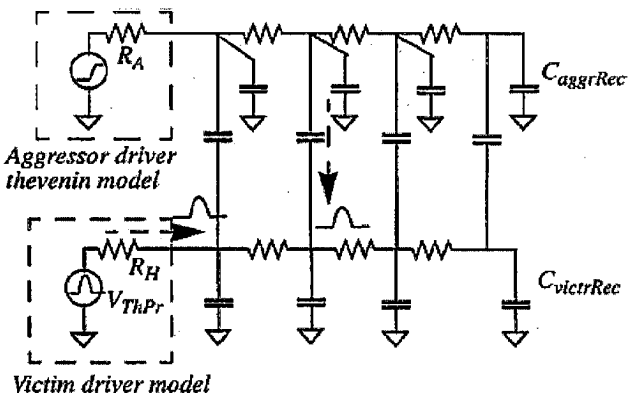


Figure 35 Linear model of noise cluster [11]

5.4 Non-linear driver model

The non-linear model of the noise cluster is depicted in Fig.36. Transforming the Thevenin models of the aggressor drivers into Norton equivalents, the nodal equations for the noise cluster can be written as follows:

$$\left(C.\frac{d}{dt}+G\right).V = J \qquad (2)$$

where C is the capacitance matrix, G is the conductance matrix, V is the vector of nodal voltages, and J is the vector of current sources. All the equations are linear except the one with the victim driver output current i_{out} which is non-linear and time dependent. The victim driver output current is expressed as a function of the victim input and output voltages, v_{in} and v_{out}, where the victim input voltage $v_{in}(t)$ is a function of time t.

$$i_{out} = f_{load}(v_{in}, v_{out}) \qquad v_{in} = f_{noise}(t) \tag{3}$$

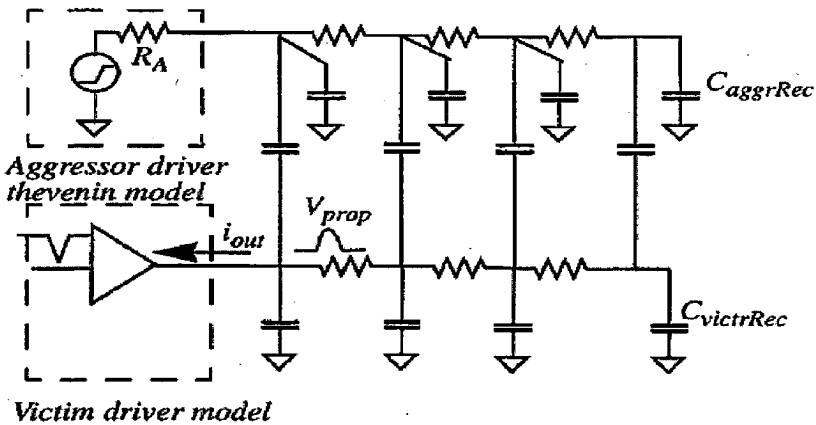


Figure 36 Non-linear model of noise cluster [11]

Since solving equations (2) and (3) simultaneously as a non-linear system is very expensive and takes away the benefits of the linear models such as superposition and model order reduction, equations (2) and (3) are solved separately and iteratively improving the solution.

5.5 Simplified non linear victim driver model

The previous approach assumes that we can compute victim output current $i_{out}(t)$ as function of time. Of course it can be done by transient simulation of the victim driver but it is slow. So this model proposes to compute it approximately but more efficiently.

Let us consider the victim drive gate as a black box described by its direct current (DC) characteristics. That is, the victim driver is modelled as a nonlinear current source (Fig. 37 (a)). Its current i_{outDC} depends on both its input and output voltages: $i_{outDC} = f_{load}(v_{in}, v_{out})$. Graphically this dependence is represented by a family of gate load curves. For our noise computations, victim driver output current $i_{outDC} = f_{load}(v_{in}, v_{out})$ is represented by a

two dimensional table and computed by DC simulations sweeping a range of input and output voltages. Also, we use DC victim driver voltage transfer characteristic expressing its output voltage as a function of the input voltage: $v_{out} = v_{out}(v_{in})$. This is given in a one dimensional table and is also computed by DC simulations sweeping the input voltage range. An example of the DC transfer curve is shown in Fig. 37(b).

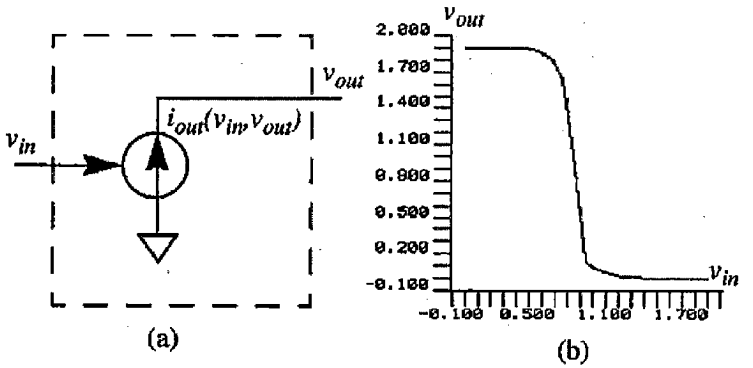


Figure 37 (a) non-linear victim driver model (b) victim driver transfer curve [11]

The family of load curves and the transfer curve are computed only once at the precharacterization stage and do not require recomputing at the time of noise analysis. Using input and output noise waveforms $v_{in}(t)$, $v_{out}(t)$ and DC characteristics of the victim driver we compute $i_{out}(t)$.

5.6 Model to determine injected noise peak

Node O corresponds to a quiet line, and represents a neighboring switching line. The attenuation along the interconnect is neglected (line lengths of a few mm or less). The resistance R_1 is the driver resistance of the aggressor (typically a few kiloohms for CMOS and a few hundreds of ohms in ECL), and C_1 is the total line capacitance of the aggressor net. X is the coupling capacitance and is proportional to the overlap length.

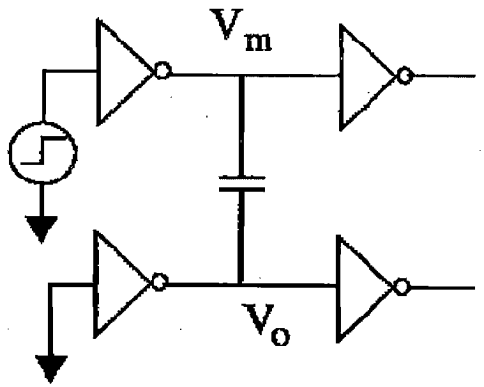


Figure 38 victim aggressor pair [22]

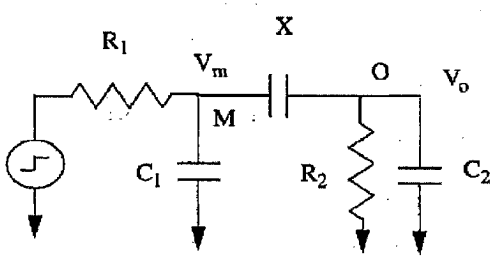


Figure 39 Equivalent circuit for computing coupled noise amplitude and width [22]

R_2 is the output resistance (typically a few kiloohms in CMOS and a few hundreds of ohms in ECL) of the victim net, and C_2 is the capacitance to ground of the victim net.

The nodal equation for output node O is

$$(X + C_2).\frac{dv_0}{dt} + \frac{v_0}{R_2} - X.\frac{dv_m}{dt} = 0 \qquad (3)$$

When the coupled noise pulse is at its maximum, the derivative of v_0 with respect to time is zero. Using this in (3), we get

$$\frac{V_P}{R_2} = \left(X.\frac{dv_m}{dt}\right)_{v_0=V_P}$$

Hence the peak of glitch is

$$V_P = R_2 \cdot \left(X \cdot \frac{dv_m}{dt} \right)_{v_0 = V_P}$$

5.7 Proposed model

The previous model only determines the injected noise peak and neglects the propagated noise.

A new model is proposed to take into account the propagated noise and gives the total noise peak after combining both injected and propagated noise.

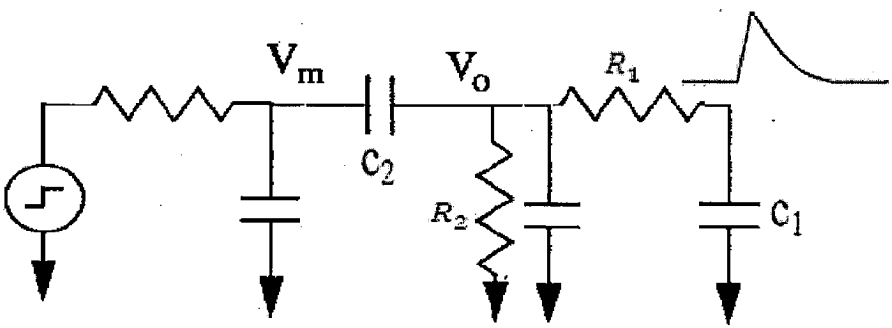


Figure 40 Proposed model

Using the nodal analysis and derivative at the peak approach (as in the previous model) we get the combined peak as

$$V_P = R_2 \cdot \left(C_2 \frac{dv_m}{dt} + C_1 \frac{dv_{np}}{dt} \right)$$

Where v_m is the injected noise and v_{np} is the propagated noise.

5.8 Results

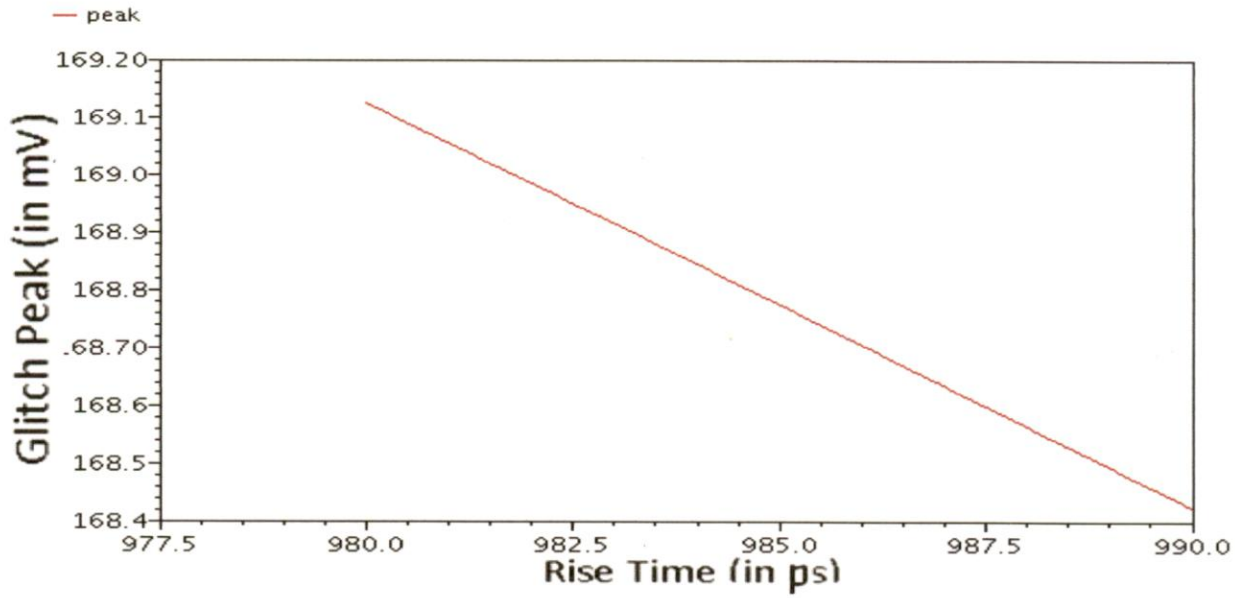


Figure 41 Propagated peak variation with slew rate for load capacitance 20fF

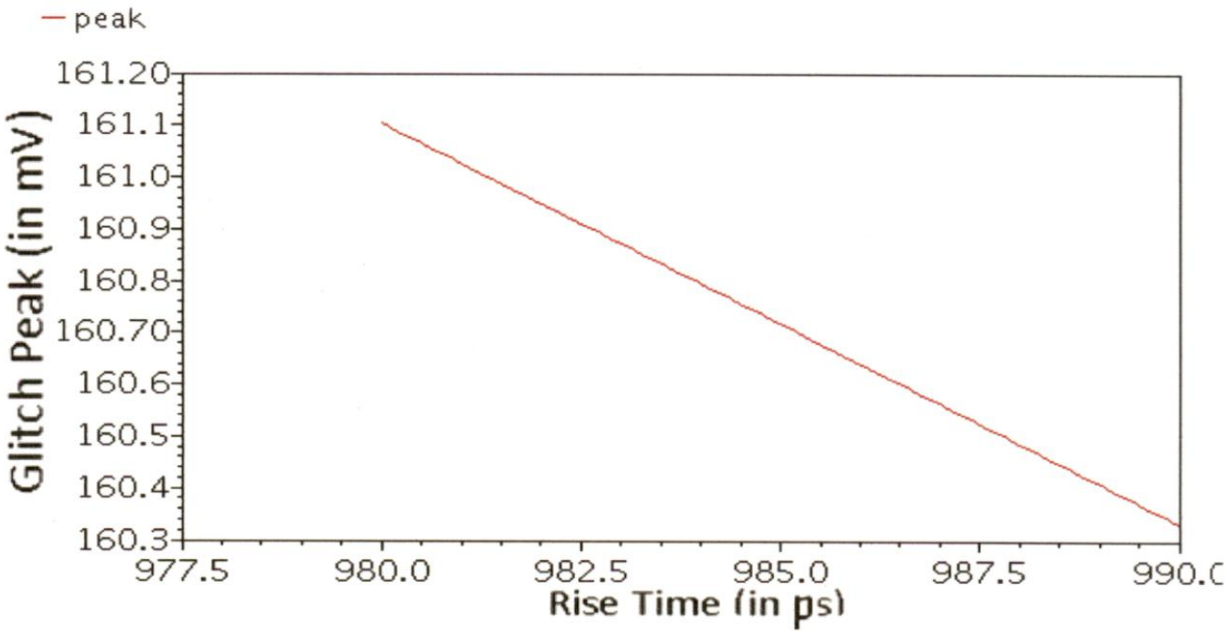


Figure 42 Propagated peak variation with slew rate for load capacitance 30fF

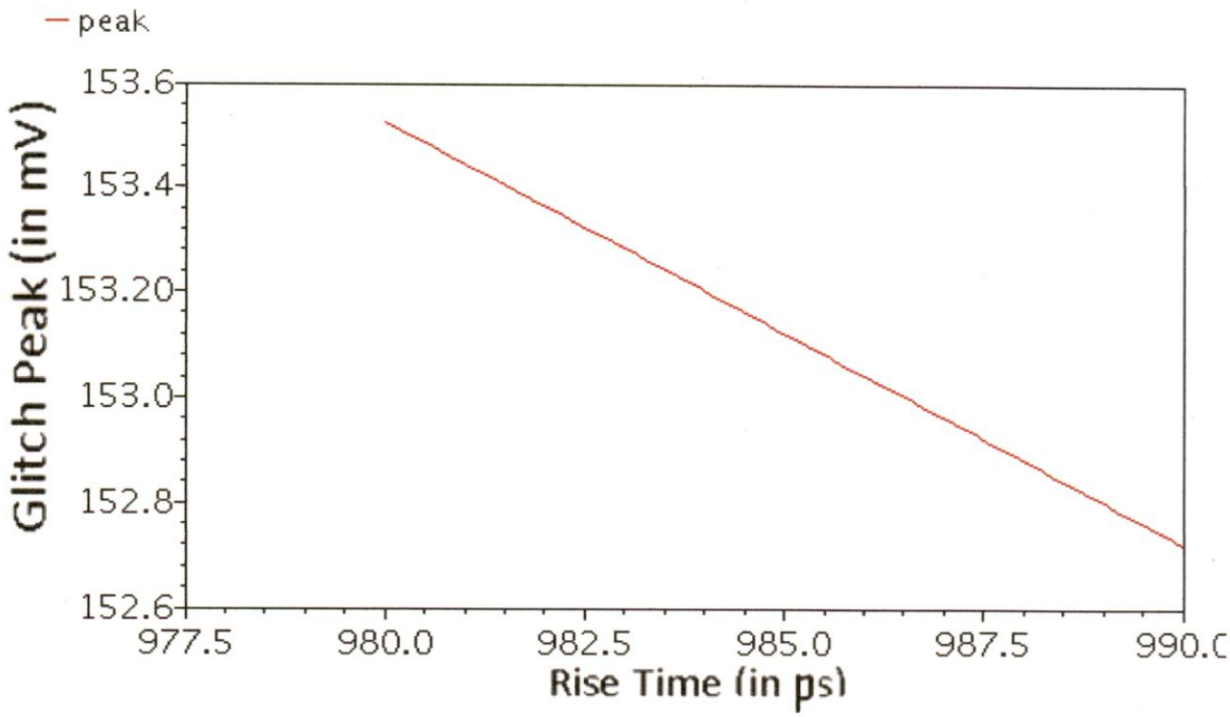


Figure 43 Propagated peak variation with slew rate for load capacitance 40f

The linear variation of propagated noise with slew rate (keeping injected noise zero) validates the linear superposition approach.

6. Delay noise analysis

6.1 Introduction

Interconnect noise can have a significant impact on gate-level timing. Noise caused by interconnect effects causes *dynamic delay*, which refers to the uncertainty in delay of a stage (gate + wire) due to the switching activity of nearby gates. For static CMOS designs, the functional implications of crosstalk aren't as significant as the potential timing errors caused by dynamic delay. Due to the restoring nature of CMOS logic, a noise glitch would need to exceed the fan-out gate's switching threshold in order to cause functional failure. This switching threshold is normally close to half of V_{DD}. In contrast, the delay change resulting from dynamic delay can easily exceed 20-30% for relatively short wires (< 0.5 mm), depending on driver and interconnect configurations. This degree of delay uncertainty is intolerable for designs with tight timing budgets [18].

It is also apparent that the magnitude of noise, both glitch and dynamic delay, are increasing with CMOS technology scaling. To first order, the amount of dynamic delay is proportional to the ratio of coupling capacitance (Cc) to total stage capacitance (including junction, fan-out, and interconnect ground capacitances). The portion of interconnect capacitance attributable to coupling has risen to about 80% for minimum-pitch wiring, both global and local. If we assume that, for global wiring, interconnect capacitance dominates gate loading, the amount of dynamic delay can reach ±80% of the nominal delay. To illustrate, Figure 44 shows the increase in delay uncertainty for a 3 mm global wire through a number of technology generations. A large inverter with fan-out of 1 serves as both victim and aggressor. The worst-case dynamic delay approaches the 80% plateau, corresponding to the portion of capacitance due to coupling. Since there is only a single aggressor in these simulations, the delay will only fluctuate approximately 80/2, or 40%, above or below the nominal delay value. Also, the low-to-high transition experiences more dynamic delay since the PMOS victim device pulling up in this scenario is weaker than the NMOS aggressor.

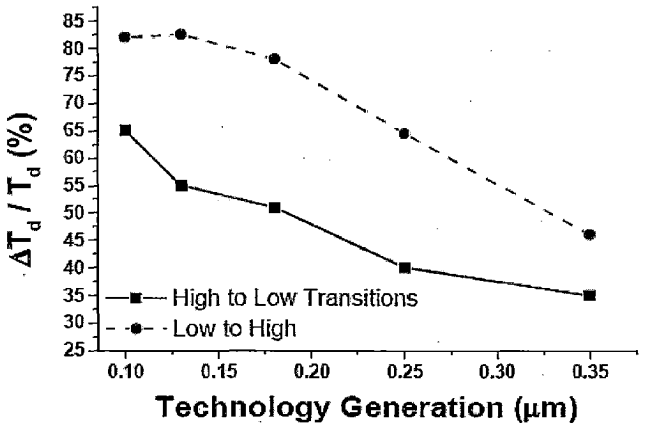


Figure 44 Technology scaling leading to dynamic delay effects

6.2 Modelling of dynamic delay

There are two primary modeling approaches to dynamic delay. The first is based on the Miller effect, which replaces a capacitance between two nodes by equivalent capacitances to ground from each node. In an on-chip context, the coupling capacitance between two adjacent wires is replaced by a ground capacitance for each net. The resulting ground capacitance has traditionally been set to either 0 or 2*Cc which have long been considered lower and upper bounds respectively. Recent work has shown that the actual bounds on the effective coupling capacitance are –1*Cc and 3*Cc. Traditionally, this approach is limited to cases where the victim and aggressor configurations are very similar – their rise times or driver strengths needed to be almost identical for the switch factor to yield accurate results.

The second modeling approach to dynamic delay recognizes the fundamental relationship between crosstalk and dynamic delay. The neighboring wires can be viewed as an added load for the victim gate and as such, we should be able to directly calculate the additional charge required to switch these new loads. By examining the voltage glitch experienced on the victim line in the crosstalk scenario, we can find an upper bound on the amount of charge needed to counteract the influence of the aggressors. In short, dynamic delay can be characterized by superimposing the voltage glitch of the victim

experiencing crosstalk onto the switching waveform of the victim when aggressors are quiet. While not exact (due to device non-linearities), this approach has been shown to yield good results for a variety of driver and interconnect dimensions.

Recently there have been several approaches presented which incorporate noise in static timing analysis (STA). All of these methods are based on the use of switching windows; each stage has a possible window of time in which it can switch state. If the switching windows of coupled nets overlap, there is the potential for dynamic delay effects. One problem that researchers have to overcome is that the size of the switching windows actually depends on the presence of noise effects. Solutions to this chicken-and-egg problem often resort to iterations and/or pessimistic initial estimates of the noise problem.

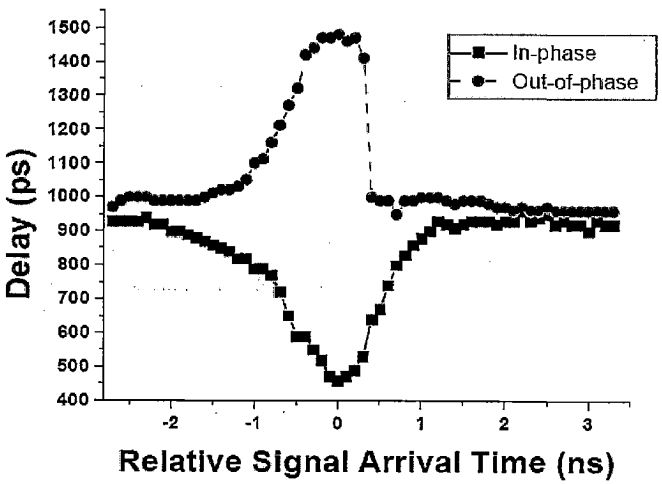


Figure 45 Measured delay change curve (DCC)

The difference of switching time between the aggressor and the victim is referred to as ***skew*** [7]. The ***input skew*** is a difference of switching time at 50% point of victim's and aggressor's input waveforms. The ***starting point skew*** is a difference between the switching starting points of the victim's and aggressor's input waveforms. The ***output***

skew is a difference at 50% point of victim's output (Out1 in Figure 46) and aggressor's output (Out2 in Figure 46) waveform.

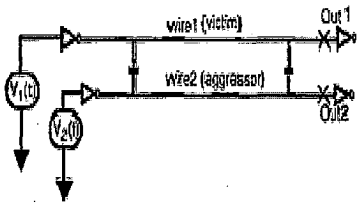


Figure 46 Victim aggressor model

When aggressor and victim switch in the opposite directions, the victim may be slowed down compared to the case when the aggressor is quiet (nominal case). When aggressor and victim switch in the same direction, the victim may be sped up. How big the slow down or speed up are at the victim's output depends on the skew and slew rates of victim's and aggressor's inputs.

6.3 Results of delay noise analysis

Delay noise analysis was done by implementing the model of Fig 46 in a 90 nm design. For both victim and aggressor switching in the same direction, the variation of delay was plotted with varying input skew, aggressor slew and coupling capacitance. The same was repeated with both nets switching in opposite directions and the following trends were observed.

Delay remains constant outside a timing window and it decreases to a minimum value inside that window near the point when the input skew is nearly zero for both the nets switching in the same direction. For both the nets switching in the same direction it increases to a maximum value inside that window near the point when the input skew is nearly zero.

For both the nets switching in the same direction as we vary the coupling capacitance delay decreases parabolically but it increases linearly for both the nets switching in opposite direction.

For both the nets switching in the same direction as we vary the aggressor slew, delay increases linearly. But for both the nets switching in the same direction it increases linearly upto the point of same slew as the victim and then shows a drastic fall.

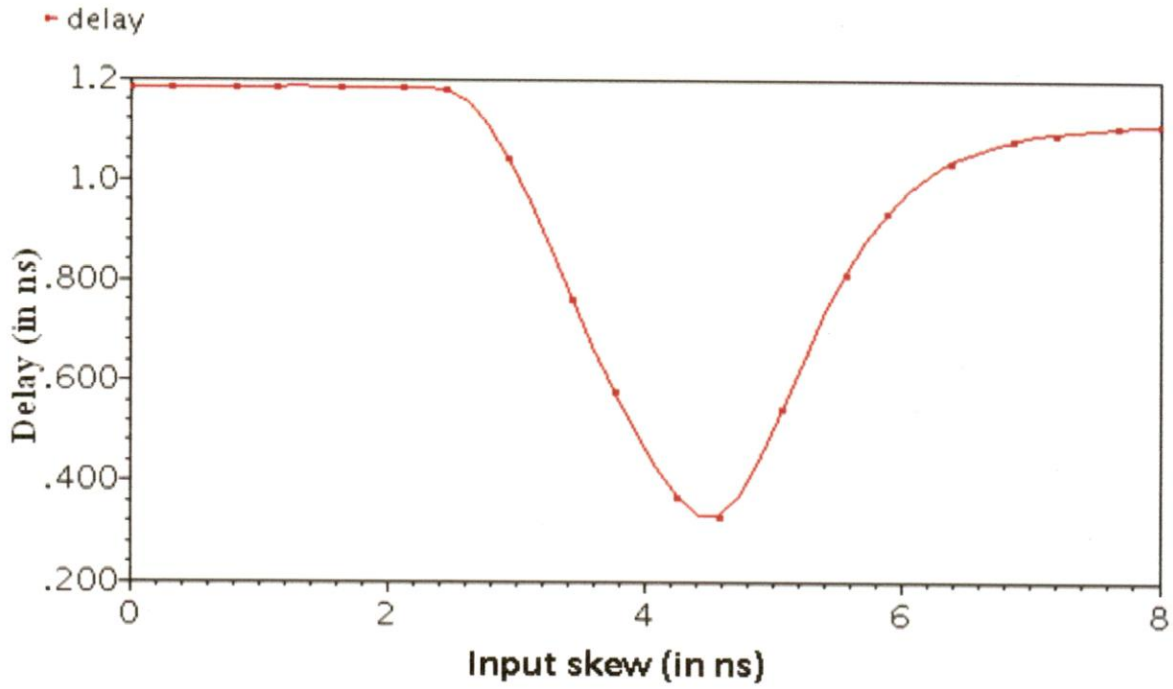


Figure 47 Delay vs skew with both nets rising

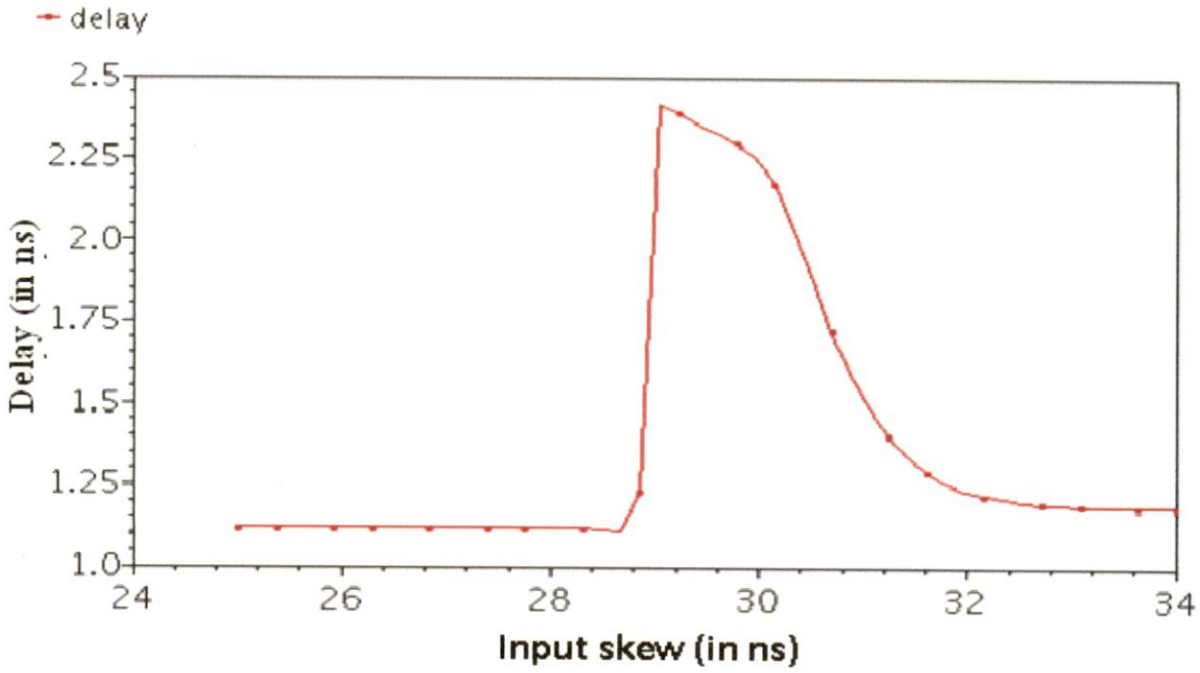


Figure 48 Delay vs skew with victim rising aggressor falling

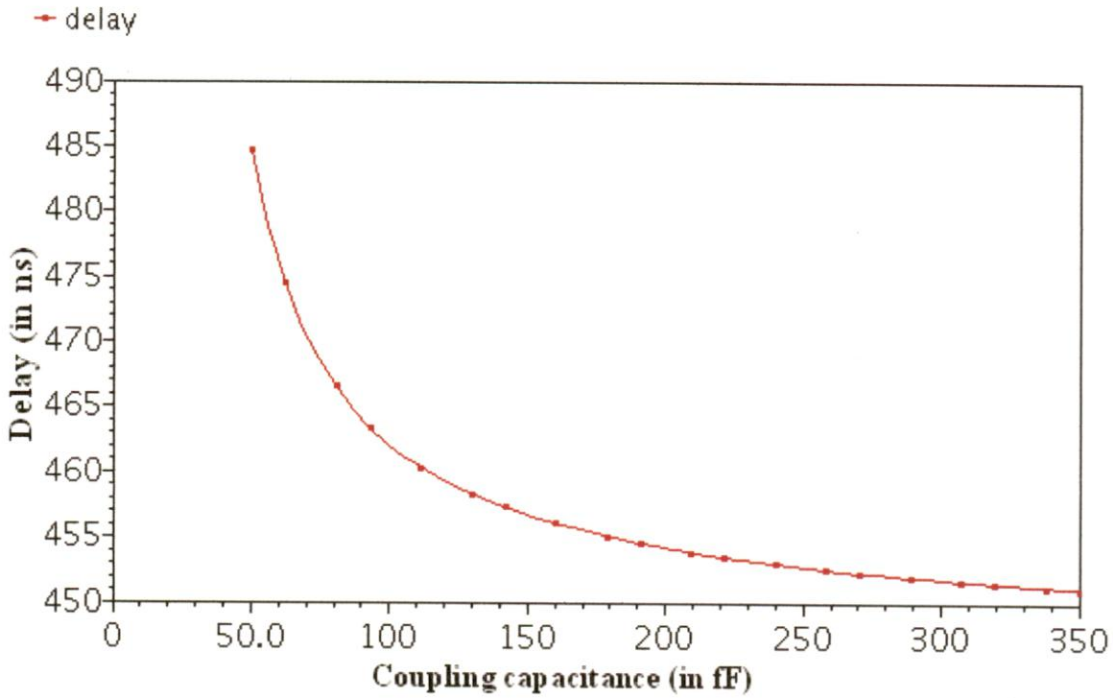


Figure 49 Delay vs coupling capacitance with both nets rising

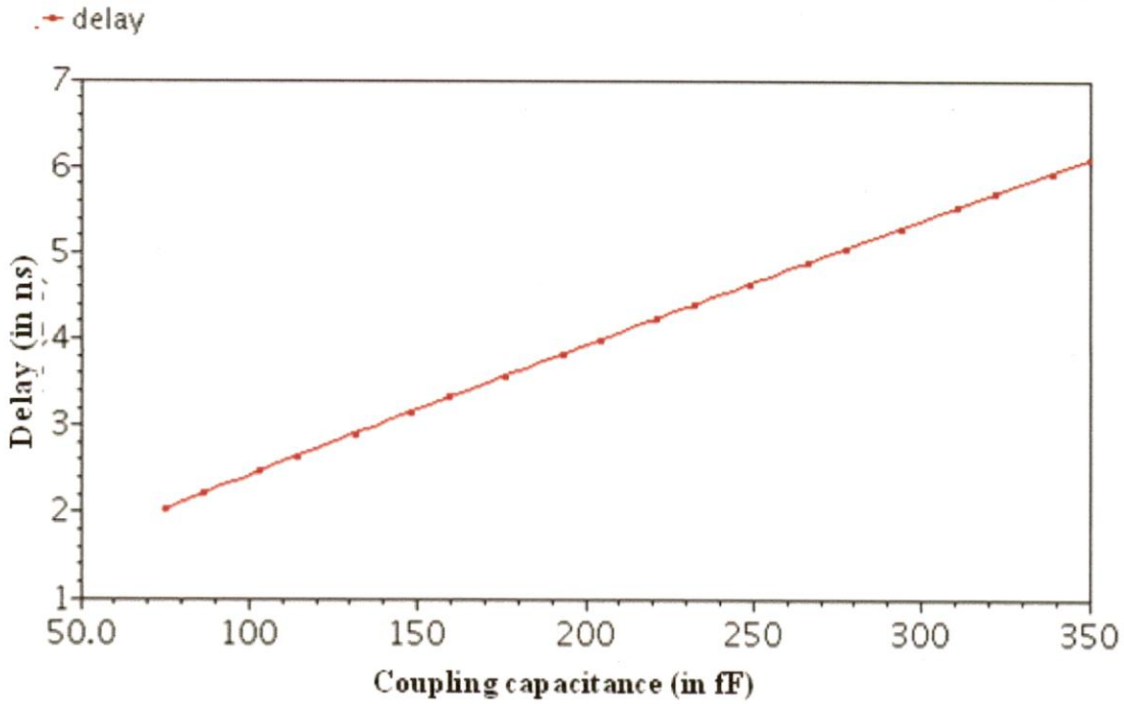


Figure 50 Delay vs coupling capacitance with victim rising aggressor falling

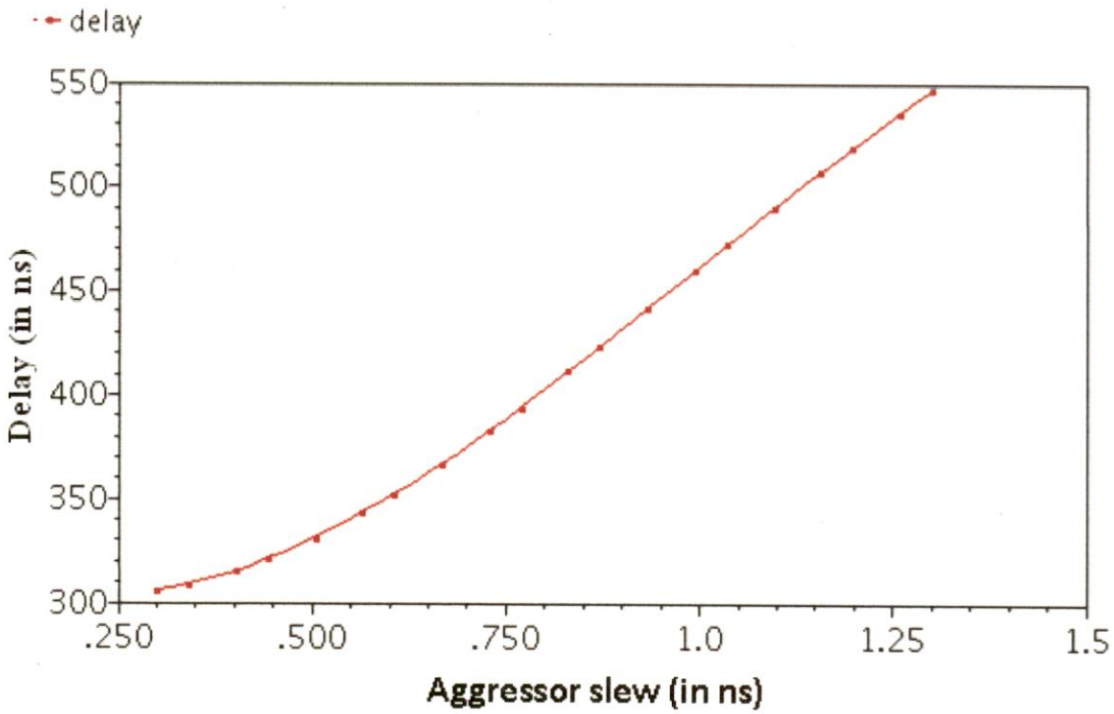


Figure 51 Delay vs aggressor slew with both nets rising

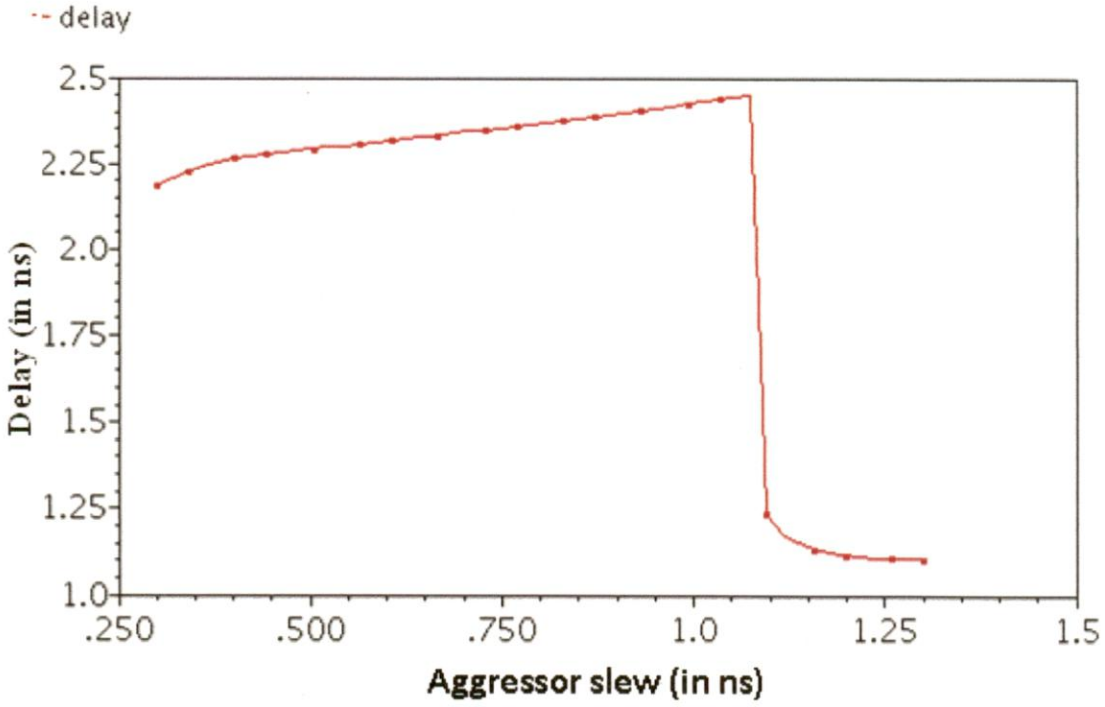


Figure 52 Delay vs aggressor slew with victim rising aggressor falling

6.4 Statistical Analysis of Delay noise

Timing verification has been moving away from the traditional over-pessimistic best/worst case analysis and addresses increased variability. Traditional timing analysis takes into account only die to-die variations, by computing minimum and maximum delays separately, and verifying timing requirements between either minimum or maximum path delays. Corner based timing analysis takes into account on-chip variation by computing minimum and maximum delays simultaneously, and allowing timing verification between minimum and maximum path delays. Statistical static timing analysis (SSTA) computes delay distribution for each pin (block based) or path (path-based), and provides "timing yield" or probability for a chip to meet its timing requirements.

Gate delay variation also comes from (1) input signal transition time variation, (2) variation of process parameters, e.g., channel length and threshold voltage of a transistor, and (3) supply voltage variation and multiple-input switching effect. To combine the effects of multiple correlated variations, an effective approach is to (1) reduce the number of variational variables, e.g., via PCA (2) represent delays in closed form functions of the variational variables (3) compute signal arrival times, and (4) achieve timing distribution by sampling and regression based on the correlations between the variational variables.

For independent variations, e.g., crosstalk alignment and gate length variations, we achieve better efficiency by computing the driver gate delay variation by superposition.

6.5 Results of Monte Carlo Simulation

Monte Carlo simulation was done by varying the input skew, aggressor slew and V_{th0} with different variance values and the variation of delay was plotted with individual variation and combined variations.

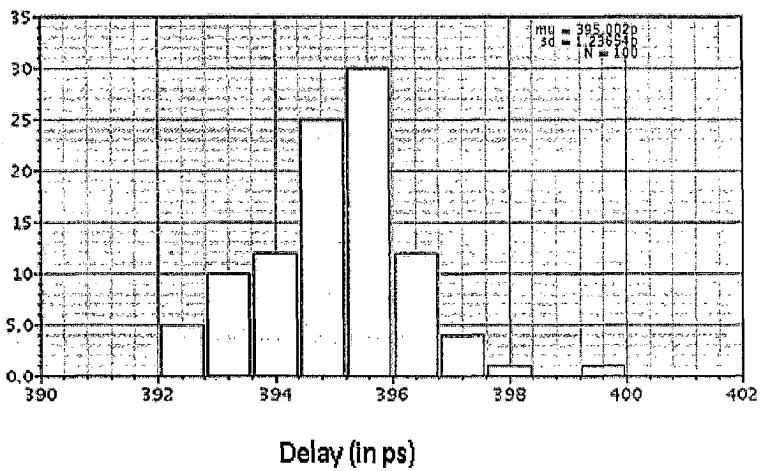


Figure 53 Delay vs skew variance 5%

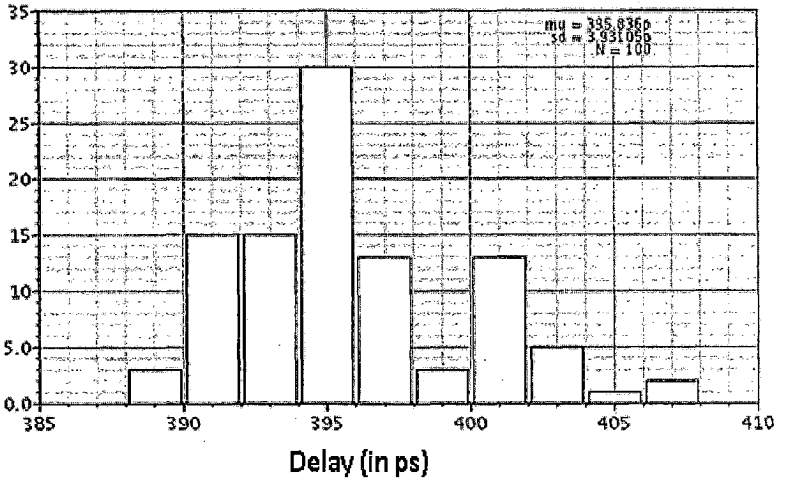


Figure 54 Delay vs skew variance 15 %

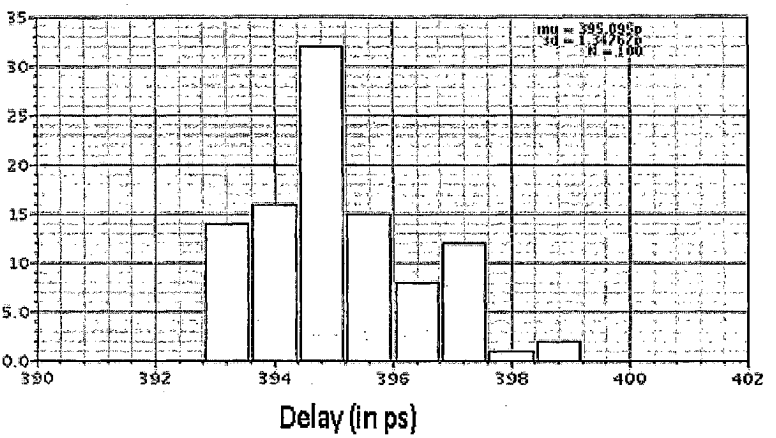


Figure 55 Delay vs slew variance 5 %

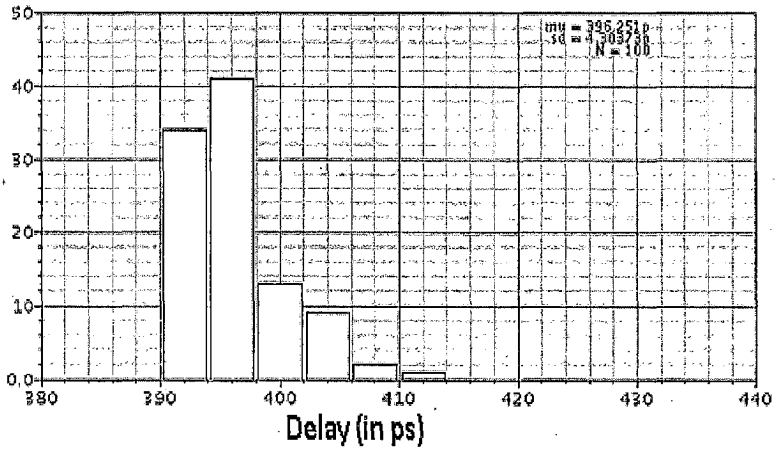


Figure 56 Delay vs slew variance 15 %

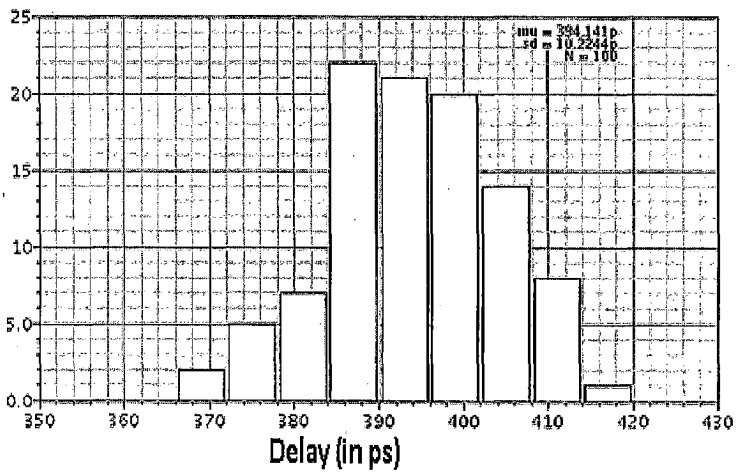


Figure 57 Delay vs V_{th0} variance 5 %

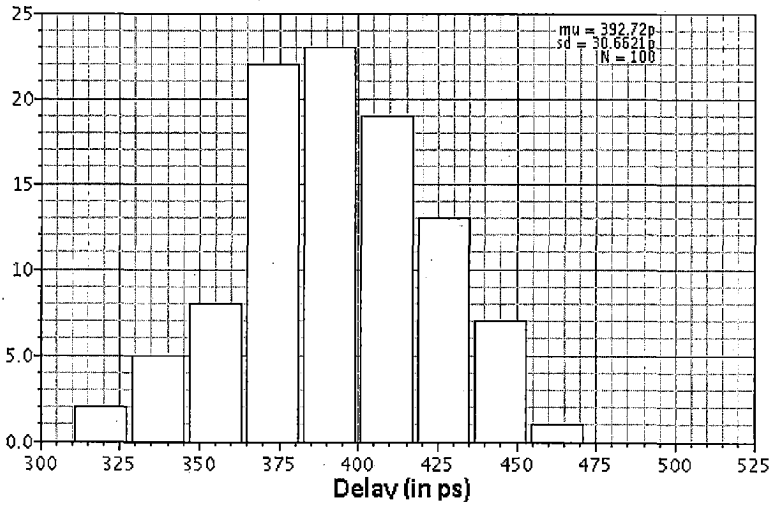


Figure 58 Delay vs V_{th0} variance 15 %

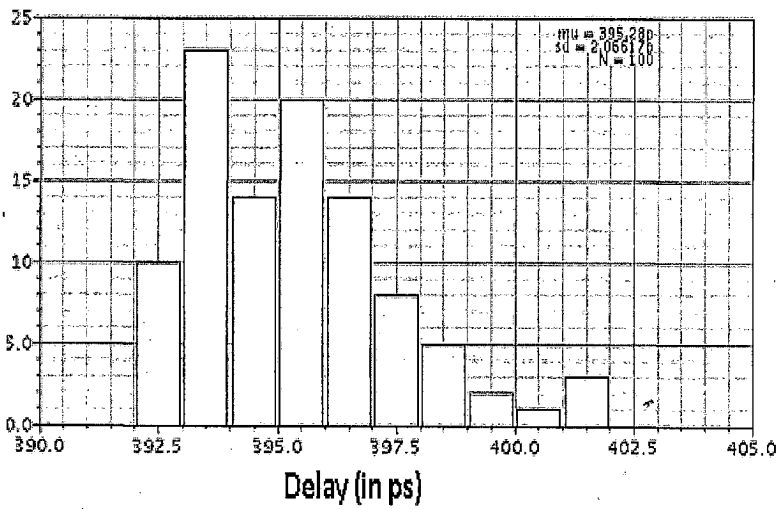


Figure 59 Delay vs skew and slew variance 5 %

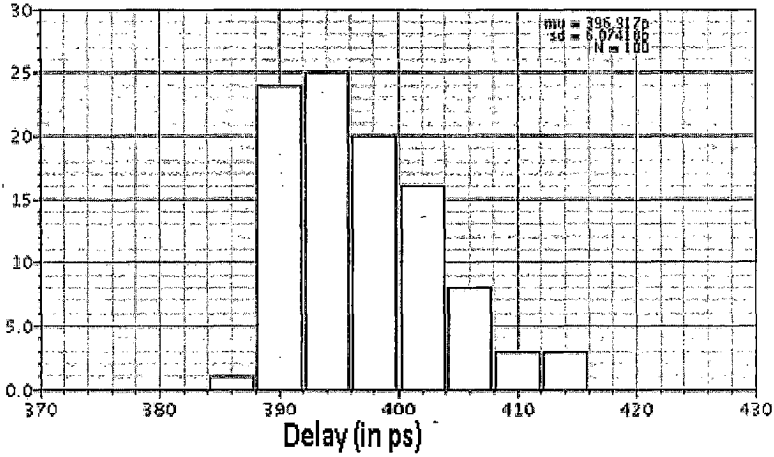


Figure 60 Delay vs skew and slew variance 15 %

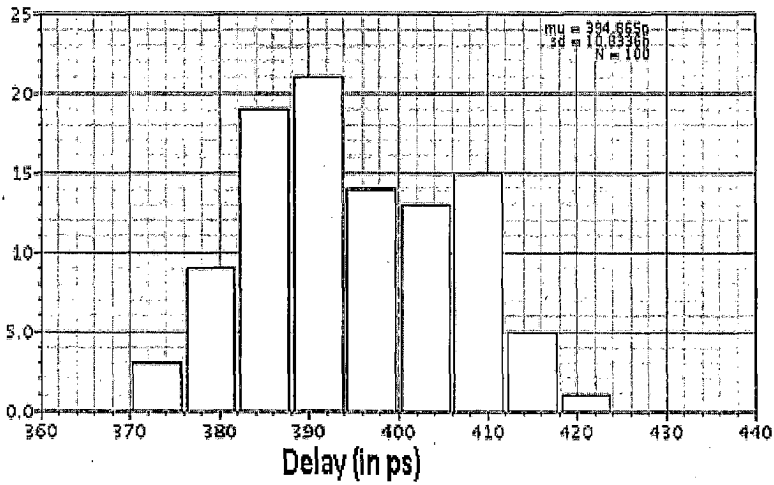


Figure 61 Delay vs skew, slew and V_{th0} variance 5 %

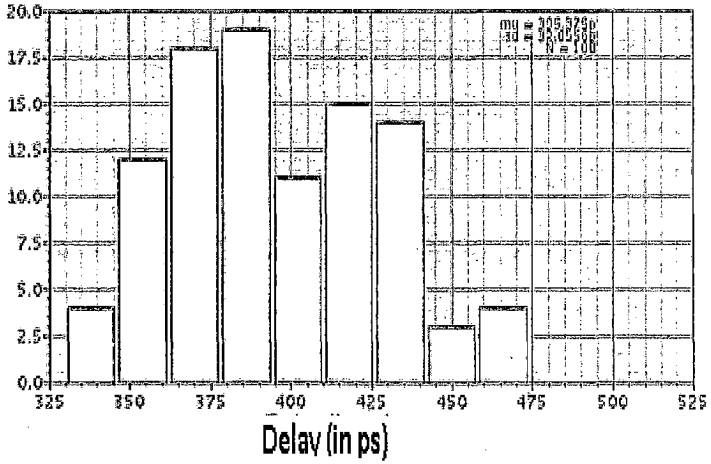


Figure 62 Delay vs skew slew and V_{th0} variance 15 %

Table 1 Delay variation with variation in input skew, aggressor slew and V_{th0}

	Delay with Input skew variation		Delay with Aggressor slew variation		Delay with V_{th0} variation		Delay with Input skew and aggressor slew variation		Delay with Input skew aggressor slew and V_{th0} variation		Avg. delay	Error %
σ^2	μ_d (p)	σ_d (p)	μ_d (p)	σ_d (p)	μ_d (p)	σ_d (p)	μ_d (p)	σ_d (p)	μ_d (p)	σ_d (p)		
5%	395.002	1.236	395.095	1.347	394.14	10.224	395.28	2.066	394.865	10.8336	394.745	0.03
15%	395.836	3.931	396.25	4.303	392.72	30.66	396.917	6.074	395.375	32.055	394.935	0.11

7. Conclusion

In this report, an SSTA like approach was followed for analysis of functional noise and delay noise due to crosstalk. A framework was built for statistical characterization of noise libraries of logic gates thus making the characterization approach suitable for taking into account the effects of process variations in crosstalk noise analysis.

Glitch area, being an important parameter of functional noise, has to be considered while checking a gate for noise rejection. Its linear variation with process parameters simplifies the statistical approach to crosstalk analysis.

A new hybrid model based on the triangular and 2-pole model was proposed for the coupling glitch to take into account the effect of process variations on the width of the glitch thus characterizing the noise rejection curve indirectly as a function of process parameters.

Also, a new model was proposed for the aggressor-victim pair to take into account the propagated noise. The combined peak of injected noise and propagated noise was derived and simulations were performed to validate the linear superposition approach while adding the injected noise and propagated noise.

8. References

[1] Becer, M., Vaidyanathan, R., Oh, C., and Panda, R.: "Crosstalk noise control in an SoC physical design flow", IEEE Trans. Comput. Aided Des. Integr. Circuits Syst., 23, (4), pp. 488–497, 2004.

[2] Shepard, K. L., Narayanan, V., "Conquering Noise in Deep Submicron Digital ICs." IEEE Trans. on Design & Test for Computers, vol.15, no.1, pp.51-62, Jan. 1998.

[3] John W. Poulton, William J. Dally, "Digital Systems Engineering", Cambridge University Press,1998.

[4] Shepard, K. L., Narayanan, V., "Noise in Deep SubMicron Digital Design", Proceedings of ICCAD, pp. 524-531,1996.

[5] Shepard, K. L., "Design Methodologies for Noise in Digital Integrated Circuits", Proceedings of IEEE Design Automation Conference, pp.94-99, 1998.

[6] S. Alwar, D. Blaauw, A. Dasgupta, A. Grinshpon, R. Levy, C. Oh, B. Orshaw, S. Sirichotiyakul, and V . Zolotov, "Clarinet: A Noise Analysis Tool for Deep Submicron Design," in Proceedings of Design Automation Conference DAC, pp. 233-238, June 2000.

[7] Shepard, K. L., Narayanan, V., Rose, R.," Harmony: Static Noise Analysis of Deep Sub Micron Digital Integrated Circuits", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol.18, no.8, pp.1132-1150, Aug 1999.

[8] J.M.Rabaey, A.Chandrakasan, B. Nikolic, "Digital Integrated Circuits", Prentice Hall, 2003

[9] S. Chandrasekar, G. Varshney, and V. Visvanathan, "A comprehensive methodology

for noise characterization of ASIC cell libraries," *Proc. of Intl. Symp. on Quality of Electronic Design*, pp.530 – 535, 2005.

[10] K. L. Shepard and K. Chou, "Cell characterization for noise stability", IEEE CICC, pp. 91-94, May 2000.

[11] Zolotov, V., Blaauw, D., Sirichotiyakul, S., Becer, M., Oh, C., Panda, R., Grinshpon, A., Levy, R. "Noise Propagation and Failure Criteria for VLSI Designs." ICCAD pp.531 – 538, 2002.

[12] Alireza Kasnavi, Joddy W. Wang, Mahmoud Shahram, and Jindrich Zejda, "Analytical Modeling of Crosstalk Noise Waveforms with Four-Parameter Weibull Function", ICCAD, vol., no., pp. 141-146, 7-11 Nov. 2004.

[13] Wei-yu Chen, Sandeep K. Gupta and Melvin A. Breuer, "Analytical Model for Crosstalk Excitation and Propagation in VLSI Circuits", IEEE Transactions of Computer-Aided Design of Integration Circuits and Systems, Vol. 21, No. 10, pp. 540-547, Oct.2002.

[14] Shrivastava, S.; Parameswaran, H., "Statistical Crosstalk Noise Analysis Using First Order Parameterized Approach for Aggressor Grouping," *ISQED,* vol., no., pp.445-449, March 2008.

[15]H.Chang and S.Sapatnekar, "Statistical Timing Analysis Under Spatial Correlations", *IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS*, Vol. 24, No. 9, pp.1467-1482, September 2005.

[16] Ratnakar Goyal, Sachin Shrivastava, Harindranath Parameswaran and Parveen Khurana,"Improved First-Order Parameterized Statistical Timing Analysis for Handling Slew and Capacitance Variation", *20th IEEE Conference on VLSI Design*, pp.932 – 939, 2007.

[17] Lou Scheffer, "Explicit Computation of Performance as a Function of Process Variation", *Proc. TAU*, pp.630 – 636, 2002.

[18] Sirichotiyakul, S., Blaauw, D., Oh, C., Levy, R., Zolotov, V., Zuo, J., "Driver Modeling and Alignment for Worst-Case Delay Noise", DAC, pp.425-429, 2001.

[19] A.Korshak, "Noise-rejection model based on charge-transfer equation for digital CMOS circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* vol.23, no. 10, pp. 1460-1468, Oct 2004.

[20] A. B. Kahng , B. Liu and X. Xu "Statistical gate delay calculation with crosstalk alignment consideration"," *Proc. Great Lakes Symp. VLSI*, pp. 223-230, 2006.

[21] Tseng,K., Kariat, V., "Static Noise Analysis with Noise Windows" , DAC, pp. 864-870, 2003.

[22] A.Vittal and M. Marek-Sadowska, "Crosstalk reduction for VLSI," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* vol.16,no.3, pp.290-298, Mar 1997.

[23] Mridul Agarwal, Kanak Agarwal, Dennis Sylvester, David Blaauw, "Statistical Modeling of Cross-Coupling Effects in VLSI Interconnects", *Proc. Asia and South Pacific Design Automation Conference*, pp. 503-506, 2005.

[24] A. Gyure, A. Kasnavi, S. Lo, P.F. Tehrani, W. Shu, M. Shahram, J.W. Wang, J. Zedja, "Noise library characterization for large capacity static noise analysis tools," *ISQED*, vol., no., pp. 28-34, 21-23 March 2005.

[25] Chen, P.; Kirkpatrick, D.; Keutzer, K.,"*STATIC CROSSTALK-NOISE ANALYSIS For Deep Submicron Digital Designs"*, Springer, 2004.