

STATISTICAL STATIC TIMING ANALYSIS CONSIDERING PROCESS  
VARIATIONS AND CROSSTALK

A Thesis

by

SENTHILKUMAR VELUSWAMI

Submitted to the Office of Graduate Studies of  
Texas A&M University  
in partial fulfillment of the requirements for the degree of  
MASTER OF SCIENCE

August 2005

Major Subject: Computer Science

STATISTICAL STATIC TIMING ANALYSIS CONSIDERING PROCESS  
VARIATIONS AND CROSSTALK

A Thesis

by

SENTHILKUMAR VELUSWAMI

Submitted to the Office of Graduate Studies of  
Texas A&M University  
in partial fulfillment of the requirements  
for the degree of

MASTER OF SCIENCE

Approved by:

Chair of Committee, Duncan M. (Hank) Walker  
Committee Members, Rabi Mahapatra  
Vivek Sarin  
Weiping Shi  
Head of Department, Valerie Taylor

August 2005

Major Subject: Computer Science

## ABSTRACT

Statistical Static Timing Analysis Considering Process

Variations and Crosstalk. (August 2005)

Senthilkumar Veluswami, B.E.,

Anna University, Chennai, India

Chair of Advisory Committee: Dr. Duncan Moore Henry Walker

Increasing relative semiconductor process variations are making the prediction of realistic worst-case integrated circuit delay or sign-off yield more difficult. As process geometries shrink, intra-die variations have become dominant and it is imperative to model them to obtain accurate timing analysis results. In addition, intra-die process variations are spatially correlated due to pattern dependencies in the manufacturing process. Any statistical static timing analysis (SSTA) tool is incomplete without a model for signal crosstalk, as critical path delays can increase or decrease depending on the switching of capacitively coupled nets. The coupled signal timing in turn depends on the process variations. This work describes an SSTA tool that models signal crosstalk and spatial correlation in intra-die process variations, along with gradients and inter-die variations.

DEDICATION

To my parents

## ACKNOWLEDGMENTS

First, I would like to thank Dr. Walker for advising me during the past one and a half years. I really appreciate his simple and practical approach to solving problems in the context of the industry. I have learned many things from him and consider myself fortunate to have been one of his students.

I would also like to thank Drs. Mahapatra, Sarin and Shi for being on my committee. I am grateful to Dr. Shi for allowing me to use his research framework and data in my work. I am fortunate to have attended a course on VLSI circuit modeling and optimization by Dr. Jiang Hu.

I would like to thank Xiang Lu for integrating the new models presented in this work in his research framework. He has also helped me in running some of my simulations and sorting out many implementation issues.

I would also like to thank my roommates and other friends for making my stay in College Station a memorable one. I would like to thank Elena and Patricia for taking care of all my administrative issues in the graduate office in the department.

I am grateful to Dr. Robert K. James and Dr. Craig Wilson of the Department of Teaching, Learning and Culture, and Dr. Michael K. Lindell of the Hazard Reduction and Recovery Center in the College of Architecture for providing financial support during my graduate studies.

I would like to thank my sister Sri and her husband Pavan for all their support during my stay here in US. I really enjoyed the company of Nishanth and the new entrant Esha. Last, but certainly the most, I would like to thank my parents and family for their constant support and believing in my abilities.

## TABLE OF CONTENTS

	Page
ABSTRACT.....	iii
DEDICATION.....	iv
ACKNOWLEDGMENTS .....	v
TABLE OF CONTENTS.....	vi
LIST OF FIGURES .....	viii
LIST OF TABLES.....	ix
I. INTRODUCTION .....	1
A. Timing Analysis.....	1
B. Process Variations.....	2
C. Crosstalk .....	2
D. Organization of the Thesis.....	3
II. BACKGROUND .....	4
A. Timing Analysis.....	4
B. Process Variations.....	8
C. Crosstalk .....	9
D. Testable Paths .....	15
III. SOLUTION METHODOLOGY.....	17
A. Delay Model.....	17
B. Process Variations.....	17
C. Crosstalk .....	23
D. Statistical Timing Analysis.....	28
IV. IMPLEMENTATION AND RESULTS .....	35
A. Implementation Details.....	35
B. Accuracy of SSTA Model.....	39
C. Importance of Crosstalk in SSTA.....	42
D. Effect of Different Grid Sizes on Delay Distribution.....	45
E. Correlation vs Independence.....	47
F. Validation of the Linear Crosstalk Model .....	50
V. SUMMARY AND CONCLUSION.....	55

	Page
REFERENCES .....	57
VITA.....	63

## LIST OF FIGURES

	Page
Figure 1. Normal distribution representation of delay in SSTA.....	5
Figure 2. Computing maximum delay at the output of a sample AND gate.. .....	7
Figure 3. A sample victim and aggressor in a circuit. ....	10
Figure 4. Cross-section into interconnect system with parasitic capacitances definition. ....	11
Figure 5. Grounded capacitance approach to model crosstalk. ....	13
Figure 6. Relative window method for calculating delay increase at the victim.....	14
Figure 7. Gradient example.....	18
Figure 8. 2×2 partition of a die. ....	19
Figure 9. A sample die with values in the covariance matrix for grid ‘*’ .....	21
Figure 10. Crosstalk delay increase curve. ....	25
Figure 11. Delay difference curve for piece-wise linear analysis.....	26
Figure 12. Order of maximum distribution computation.....	30
Figure 13. SSTA tool flow.....	38
Figure 14. Comparison of PDF plots of MC and $SSTA_{xtalk}$ for circuit C6288. ....	40
Figure 15. Importance of considering crosstalk in SSTA for C7552. ....	44
Figure 16. Effect of varying grid sizes for C1355. ....	47
Figure 17. Comparison of SSTA results for different correlation structures for C6288. ....	49
Figure 18. Crosstalk delay increases over relative signal arrival time in a sample circuit with the same input slew rates. ....	52
Figure 19. Crosstalk delay increases over relative signal arrival time in a sample circuit with different input slew rates. ....	53



## LIST OF TABLES

	Page
Table 1. Standard deviation of process variables.....	36
Table 2. Analysis of $SSTA_{xtalk}$ and Monte Carlo method.....	39
Table 3. Simulation time for $SSTA_{xtalk}$ analysis. ....	41
Table 4. Importance of considering crosstalk in SSTA. ....	42
Table 5. Comparison of standard deviation of SSTA results for various grid sizes	45
Table 6. Comparison of the impact of different correlation structures on SSTA with crosstalk. ....	48
Table 7. Comparison of different correlation structures without crosstalk for C1355 and C1908. ....	50

## I. INTRODUCTION

### A. *Timing Analysis*

Timing analysis is used to determine the critical (longest) delay of the circuit. The longest delay of the circuit limits the clock frequency of the circuit. Static timing analysis is attractive to circuit designers as the circuit can be analyzed quickly without simulating the circuit for every combination of primary inputs. In Static Timing Analysis, the delays are treated as constants.

As deep submicron (DSM) semiconductor technology advances, there is increasing relative uncertainty in process parameters. This makes it increasingly difficult to predict integrated circuit timing behavior [1, 2, 3]. Most of the current Static Timing Analysis (STA) tools are corner based, i.e. they approximate the maximum deviation in delay in each process corner and then calculate the worst-case delay as the sum of nominal delay and maximum delay deviations in each process parameter. This approach is too pessimistic as it assumes that the worst-case delay occurs under maximum delay deviation in all process parameters simultaneously.

An alternative approach to overcome this problem is Statistical Static Timing Analysis, in which the delays are treated as probability density functions. This approach is viable as the delays are no longer fixed numbers and have both independent and correlated components.

### *B. Process Variations*

As the feature size decreases, the influence of process variations on circuit design and performance is increasing manifold. Process variations can be classified into inter-die process variations and intra-die variations. Process parameters that change from die to die are called inter-die variations while process parameters that have different values at different points on a die are called intra-die variations. It has been found that the intra-die process variations of a gate are spatially correlated with other gates found in its neighborhood. Many of the current Statistical Static Timing Analysis (SSTA) approaches have ignored spatial correlations in intra-die process variations, i.e. they have assumed that variations within a die are independent [2, 4, 5]. Some approaches have incorporated spatial correlations into their analyses [3, 6, 7]. This work models inter-die and intra-die variations and the effect of spatial correlations.

### *C. Crosstalk*

Signal crosstalk occurs due to the interference of signals in neighboring interconnects due to capacitive or inductive coupling. In this research, we will focus on capacitive coupling. Crosstalk can increase or decrease the signal delay and/or affect the signal integrity. Crosstalk can increase the delay on the net (victim) if the coupled signals have opposite transitions (aggressors), while similar transitions on the coupled nets (helpers) can reduce the victim signal delay. Multiple aggressor signals or combinations of aggressors and helpers make the process of predicting signal delay very difficult. The switching window (time interval when a signal transition occurs) of aggressors may depend on the victim's switching window (i.e. if they are aggressor and victim to each other). This situation is similar to the classical chicken and egg problem [8]. Hence, there

is uncertainty in predicting signal transitions even without any process variations. The presence of process variations aggravates the problem further, since the switching windows and coupling capacitance are a function of process parameters. The main contribution of this work is incorporating crosstalk analysis into an SSTA framework similar to [3]. A key part of this is developing a model of crosstalk that fits into the delay model and process variation model, so that the presence of crosstalk can be viewed as causing changes in the mean and variance of the delay distribution.

#### *D. Organization of the Thesis*

The thesis is organized as follows: Section II gives an introduction to the concepts used in this work. The process variation model for inter-die and intra-die variations, the crosstalk model and the statistical static timing analysis algorithm are explained in Section III. Section IV discusses implementation details and results while we conclude the thesis and point out future directions for research in Section V.

## II. BACKGROUND

### A. *Timing Analysis*

The objective in static timing analysis is to calculate the slack at the primary outputs of the circuit. The slack is calculated by subtracting the critical (longest) delay at the primary outputs of the circuit from the maximum allowable arrival time (minimum circuit timing requirements) at the primary outputs of the circuit. A circuit with positive slack indicates that the minimum circuit timing requirements are met. A negative slack indicates that signals could potentially be too late to meet the minimum timing requirements. There is a possibility of timing violation and hence the circuit needs to be redesigned. In sequential circuits, a positive slack provides an opportunity for the designer to increase the clock frequency while a negative slack requires either the clock to be slowed down or gate delays be reduced.

#### A.1. Types of Timing Analysis

Timing analysis can be classified into Static Timing Analysis (STA) and Dynamic Timing Analysis.

Static timing analysis is vectorless, i.e. timing analysis is performed without using any input vectors. A vectorless timing analysis approach gives a quick estimate of the potential longest path and the circuit slack. Static timing analysis is divided into Deterministic Static Timing Analysis (STA) and Statistical Static Timing Analysis (SSTA).

In deterministic timing analysis, the gate and interconnect delays are usually specified as constants. Sometimes, the delays are specified as min-max values.

In statistical static timing analysis, the delays are represented as probability distributions, e.g. as normal or uniform distributions [2]. The arrival times are modeled as Cumulative Density Functions (CDFs) and gate delays as Probability Density Functions (PDFs) in [2]. Figure 1 illustrates an example of representing normally distributed delay as a PDF. SUM and MAX are the two basic operations in static timing analysis and are explained in the next section. If delays are normal, the SUM can be computed exactly. The MAX of two normals is approximated as normal [3]. SSTA is becoming more attractive to designers as STA is increasingly pessimistic.

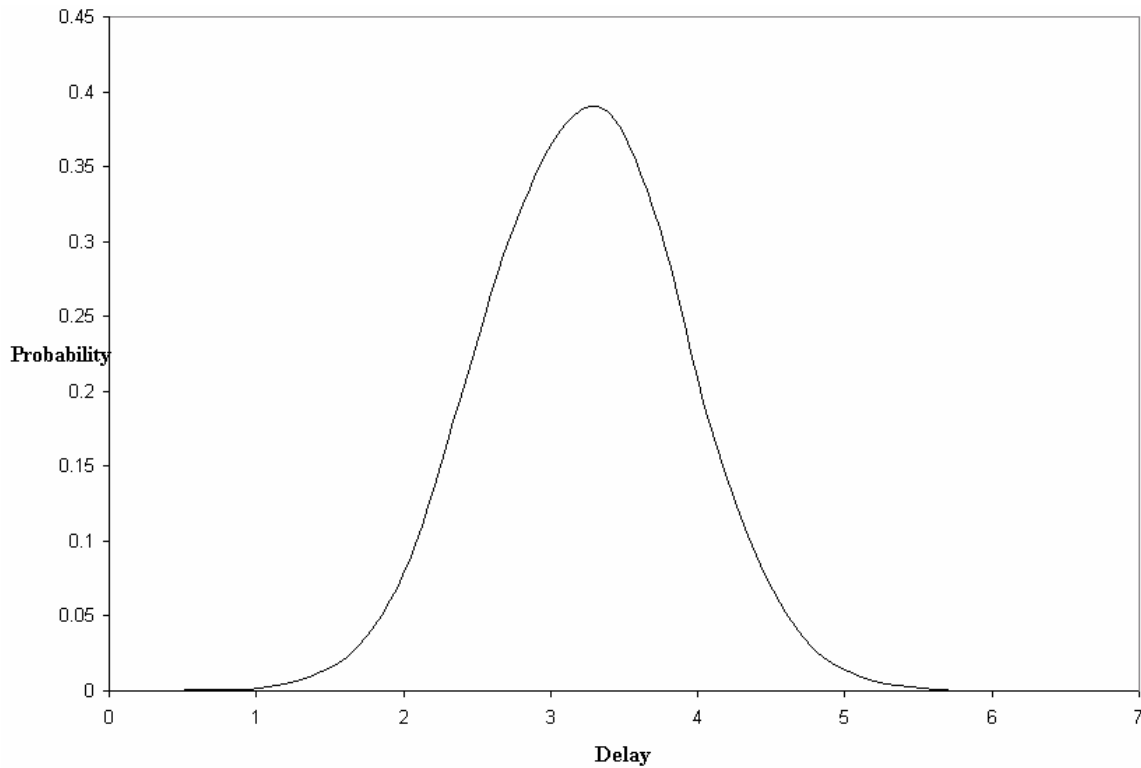


Figure 1. Normal distribution representation of delay in SSTA.

Two approaches have been followed in statistical static timing analysis. The first approach is block based [2, 3, 7] while the second approach is path based [1, 9]. In the

block-based approach, a PERT-like analysis [10] is performed on the circuit employing a SUM and MAX operation at every block or gate. In a path-based approach, a set of potentially longest paths are considered by the timing analyzer. Path based approaches have the potential to achieve higher accuracy, since the entire path is considered at once, and there is no loss in accuracy due to the approximation of the MAX operation until the results of all paths are combined. Block based approaches have the advantage that they better support incremental timing analysis [2], since they can more quickly recompute circuit delay after minor changes to the circuit structure. The trend in logic optimization is to incorporate simple statistical delay models into the synthesis procedure, rather than including a full SSTA into the loop [5].

Dynamic timing analysis is vector-based, i.e. the circuit timing is analyzed for every input vector. This approach is very costly as the number of vectors increases exponentially with the number of primary inputs, to analyze all input combinations. A circuit with 20 primary inputs can possibly have  $2^{20}$  combinations of primary inputs although many of the combinations may not occur during normal operation of the circuit. Dynamic timing analysis requires  $O(G \cdot V)$  time for a circuit with  $G$  gates and  $V$  vectors while block based STA takes only  $O(G)$  time for the same circuit.

In this work, we will focus on sign-off SSTA, where the analysis is performed once prior to fabrication, and the primary goal is to achieve high accuracy at reasonable computational cost. A key challenge in path-based STA is that its accuracy depends on the set of paths selected for analysis. Prior research has shown that 10,000 or more paths might be required to achieve high accuracy in large circuits. In SSTA, the goal is to select paths that might be the longest under some process and crosstalk condition, and provide

the tightest bounds on delay. We achieve this by generating a set of globally longest paths using a simplified version of the CodGen ATPG tool [11]. The advantage of this approach is that the amount of justification performed on the paths (i.e. eliminating false paths and finding the input vector) is within user control, so accuracy and speed can be traded. If the paths are not fully justified, then some false paths will be included which may be longer than any real paths for some process conditions, leading to predicted timing slower than actual timing. This is still more accurate than block-based analysis, which does not consider justification.

## A.2. Basic Operations in Timing Analysis

SUM and MAX are the two basic operations in static timing analysis. The arrival time at a gate input is added (through a SUM operation) to the gate delay to obtain the possible arrival time at the output of the gate through this input/fan-in. If the gate has more than one input, a MAX operation is performed on all the possible arrival times to obtain the critical (longest) delay at the output of this gate.

At circuit node  $i$ , the signal arrival time is represented by  $A_i$  while the delay from node  $i$  to another node  $j$  is represented by  $D_{ij}$ . Figure 2 shows the timing graph for a gate. The delay values are computed with Equations (1) - (3).

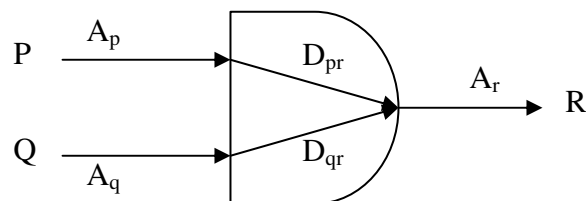


Figure 2. Computing maximum delay at the output of a sample AND gate.



$$A_{p-r} = \text{SUM} (A_p + D_{pr}) = A_p + D_{pr} \quad (1)$$

$$A_{q-r} = \text{SUM} (A_q + D_{qr}) = A_q + D_{qr} \quad (2)$$

$$A_r = \text{MAX} (A_{p-r}, A_{q-r}) \quad (3)$$

Equations (1) and (2) are examples of the SUM operation while equation (3) is an example of the MAX operation.

### B. Process Variations

Process variations can be classified into inter-die variations and intra-die variations. Inter-die variations are caused by lot-to-lot, wafer-to-wafer or across-wafer process variations. An example of across-wafer variation is radial distribution in polysilicon thickness. These die-to-die variations can be regarded as global process variations, in that they cause a variation in the mean value across the die. Intra-die variations are caused by local wafer variances, such as line width variation across a stepper field due to lens aberrations, which can lead to a process gradient across a die [3, 6], random variations such as  $V_{TH}$  variation due to dopant concentration, and pattern-dependent variations, such as line width variations due to mask pattern density. Pattern-dependent variations are deterministic, and so are assumed to be included in the parasitic extraction models, and not considered here.

#### B.1. Pelgrom Model

According to [12], the deterministic component of the process parameters is due to the device geometry while the random component of process parameters is explained by spatial correlations. The standard deviation of the difference between process parameter values in two components rises with increase in their separation distance. In other words, as the separation distance between components increases, correlation

between process parameters of the components decreases, i.e. the process parameters become more independent and hence the standard deviation of the difference between parameter values in different components becomes larger.

### *C. Crosstalk*

Crosstalk occurs due to the interference of signals in neighboring interconnects. The terms used in crosstalk analysis are described here while we discuss the source and impact of crosstalk and the various crosstalk models below. A victim net is defined as the net whose delay increases or decreases due to interference of signals from neighboring nets. An aggressor net is defined as a net which has an opposite transition with the victim net. The switching window is defined as the time interval when a signal transition may occur. If the signal of an aggressor net transitions within the switching window of the victim net and has a significant coupling capacitance with the victim net, then the delay on the victim increases. A helper net is defined as a net that switches in the same direction as the victim net. If the signal of a helper net transitions within the switching window of the victim net and has significant coupling capacitance with the victim net, then the delay on the victim decreases. A stable net is one that does not switch within the switching window. A sample victim and aggressor in a circuit are illustrated in Figure 3. The victim net is on a path that is being analyzed.

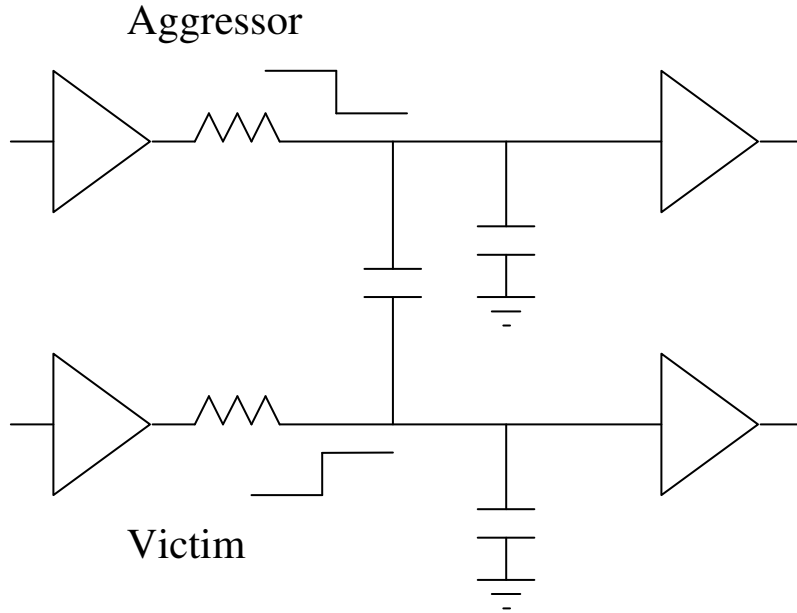


Figure 3. A sample victim and aggressor in a circuit.

Self capacitance is the value of coupling capacitance of a node to ground while cross-coupling capacitance is the value of coupling capacitance between two nodes. In deep submicron technology, the ratio of cross-coupling capacitance to self capacitance is high and is greater than 1 for many nodes [13, 14]. Figure 4 shows a cross-section of the interconnect [14]. According to [14], there is relative increase in the metal thickness ( $T$ ) with respect to metal width ( $W$ ) scaling and hence lateral or sidewall capacitance ( $C_L$ ) (which is mainly responsible for the coupling capacitance) dominates the vertical capacitance ( $C_V$ ). The aspect ratio ( $AR$ ), which is defined as the ratio of the lateral capacitance to the vertical capacitance is increasing in newer technologies and is greater than 1.

$$\frac{C_L}{C_V} = \left(\frac{T}{W}\right)^n = AR^n, n \in [1, 2]$$

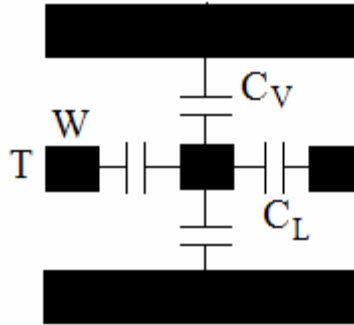


Figure 4. Cross-section into interconnect system with parasitic capacitances definition.

The relative increase in coupling capacitance means that crosstalk increasingly influences delay. The amount of delay increase or decrease depends on factors such as the switching window of the victim and the aggressors, the direction of transition at the victim, aggressor and helpers, and the relative driver strengths of victim and aggressor nets [15].

Multiple aggressor and helper signals make the process of predicting signal delay on the victim more difficult. The switching window of aggressors may depend on the victim's switching window (i.e. if they are aggressor and victim to each other). This situation can be compared to the classical chicken and egg problem [8]. Hence, there is uncertainty in predicting signal transitions even without any process variations. The presence of process variations aggravates the problem further, i.e. the switching windows are more variable than ever and hence predicting the impact of crosstalk becomes more complicated. A proper model of crosstalk that fits into the models for delay and process variations is necessary to ensure an accurate estimate of the critical delay of the circuit.

Setup time is defined as the minimum amount of time the data input at a flip-flop/latch must remain stable before the arrival of the clock signal. Hold time is defined as the minimum amount of time the data input at a flip-flop/latch must remain stable after

the arrival of the clock signal. A decrease in path delay may lead to hold violations on minimum delay paths as data may arrive too early at a latch or flip-flop. An increase in path delay may lead to setup violations on maximum delay (critical length) paths as signals may arrive too late at a latch or flip-flop. Hold violations can be corrected by making silicon changes without reducing the clock speed, but setup violations can be fixed by reducing the clock speed [16]. As we move towards deep submicron technology, the objective is to have higher clock frequencies and hence complete the complex tasks quicker than ever before. Hence, this research and a majority of researchers have focused on the potential deleterious effects of the increase in path delay due to crosstalk.

Crosstalk has been modeled in a number of ways over the last decade. One of the earliest approaches is the grounded capacitance method [17, 18, 19]. In the grounded capacitance model, the coupling capacitance ( $C_c$ ) is multiplied by the “switch factor” (SF) to obtain an equivalent grounded capacitance ( $C_{CA}$ ,  $C_{CV}$ ) as shown in Figure 5 [17]. A positive value of SF indicates an increase in victim delay while a negative value of SF indicates a decrease in victim delay. Initially, the maximum SF values were estimated at 2 but researchers found the actual SF to be more than 2 in many cases [14, 17]. Despite its inaccuracy, the grounded capacitance model is known for its efficiency in quickly estimating the delay increase due to crosstalk [20].

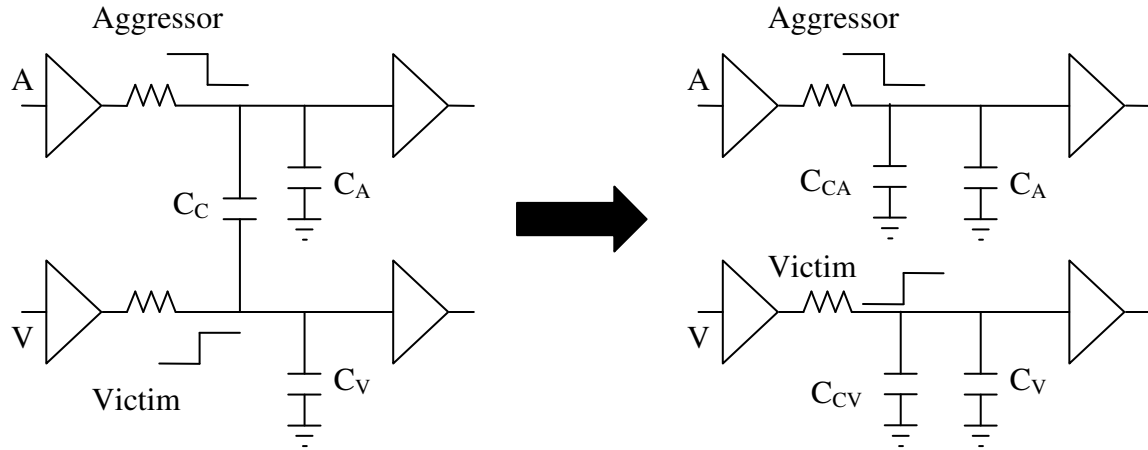


Figure 5. Grounded capacitance approach to model crosstalk.

The determination of the exact switching window of the victim is an acute problem in STA even without process variations. Some of the approaches including [21] were aimed at obtaining an aggressor alignment through an iterative procedure resulting in the worst case delay for each victim, ignoring logic constraints. This approach is akin to the corner approach in process variations where an assumption of worst case delay occurring in all the process parameters at the same time is made. Many researchers have sought to reduce this pessimism by including global timing constraints while searching for simultaneous alignments for all aggressors in the circuit [8, 22]. These approaches start with the worst case assumption of switching windows (largest switching windows) and the window shrinks with increasing number of iterations and reaches an equilibrium after which the switching windows do not shrink any further. The potential aggressors that switch within the victim's switching window are analyzed further to obtain the delay increase on the victim. This methodology has been incorporated into the STAC SSTA

[7]. But their approach ignores the logic constraints on simultaneous switching of the victim and aggressors.

An approach has been proposed that seeks to eliminate pessimism by searching for vectors that maximize crosstalk noise in combinational sub-circuits [23]. ATPG (Automatic Test Pattern Generation) techniques have been used to reduce pessimism by identifying invalid coupling interactions [24]. Gate-level logic information has also been used to eliminate invalid couplings [25].

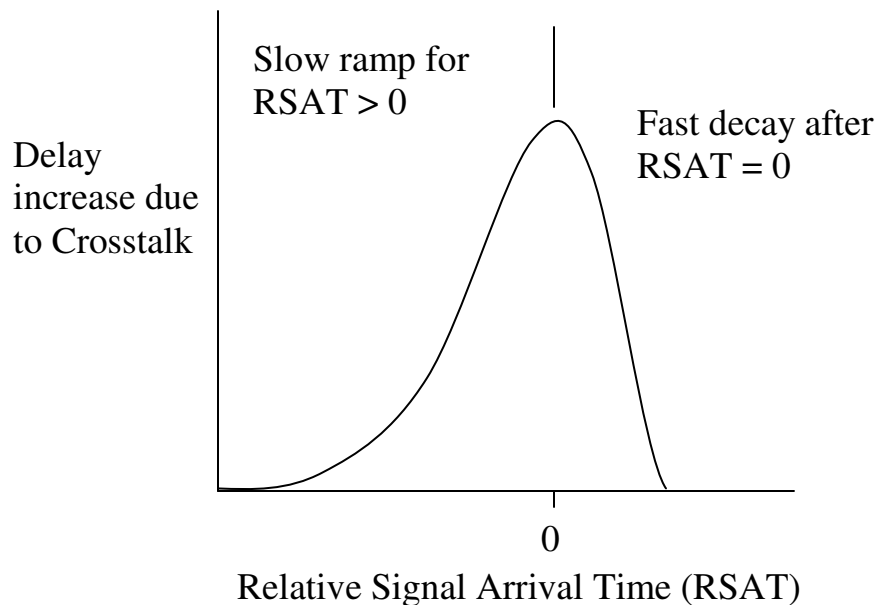


Figure 6: Relative window method for calculating delay increase at the victim.

A new approach to handle crosstalk by estimating the crosstalk delay increase as a function of the difference between victim and aggressor signal arrival times was proposed in [26]. This approach is called the “Relative Window” method. When the delay difference between the victim and the aggressor is 0, the opposite transitions (at the victim and the aggressor) switch simultaneously and there is maximum increase in delay due to crosstalk (assuming both signal transitions have the same slew rate). The increase

in victim delay decreases with the increase in the difference between the victim and aggressor arrival times as shown in Figure 6. RSAT (Relative Signal Arrival Time) is defined as the aggressor arrival time minus the victim arrival time. The curve in Figure 6 is also known as the Delay Change Curve (DCC).

An analytical method to generate the delay change curves (DCC) was introduced in [27]. A probabilistic crosstalk model that is based on the circuit topology and the short segment model (considering the effect of multiple short aggressors running in parallel to a long victim line) for a quick evaluation at the pre-layout stage was introduced in [28].

Our SSTA approach to handle the delay change due to crosstalk is based on the relative window method and the switch factor model. This approach assumes that all signal transitions have the same slew rates. The relative window method is used to estimate the delay change for different RSATs while the maximum increase in victim delay due to crosstalk (at  $RSAT = 0$ ) is calculated using the switch factor model. The switch factor used in calculating maximum delay increase due to crosstalk in SSTA is 2.

#### *D. Testable Paths*

Delay testing seeks to detect faults that make a circuit function at a lower speed than the target speed. Delay tests are classified into robust tests and non-robust tests. A robust path delay test guarantees the detection of the delay fault even if faults exist on other paths. A path which satisfies the robust path delay test criterion is called a robust path. A non-robust path delay test guarantees the detection of a path delay fault only if faults do not exist on other paths. A fault present on the off-path gates may invalidate non-robust path delay tests. A path which satisfies the non-robust path delay test criterion is called a non-robust path.



A robust path delay test criterion is more constraining than a non-robust criterion. In our path based SSTA approach, a set of longest non-robust testable paths are generated using CodGen. The set of longest non-robust testable paths are the set of ‘potentially’ longest paths under minimum path validity constraints. Paths that do not satisfy even the non-robust constraints are potential false paths which may never propagate a transition to the primary outputs. The use of false paths in SSTA may give rise to pessimistic maximum delays.

### III. SOLUTION METHODOLOGY

#### A. Delay Model

A linear model is used to approximate delay as a function of the process variables.

$$d = d_0 + s_1\Delta p_1 + s_2\Delta p_2 + \dots + s_m\Delta p_m$$

where  $d$  is the delay of a gate,  $d_0$  is its nominal delay,  $s_l$  is the delay sensitivity of process parameter  $p_l$ ,  $\Delta p_l$  is the parameter variation in  $p_l$  for this gate, and so on until  $s_m$  and  $\Delta p_m$  where  $m$  is the number of process parameters. Although delay is not exactly a linear function of the process variables, the error in approximating the delay as a linear function is small [29]. Quadratic functions achieve higher accuracy [7], but make it difficult to combine normal distributions [30].

#### B. Process Variations

Process variations are modeled as independent normally distributed random variables. Each process parameter ( $p_j$ ) is defined as follows:

$$p_j = \overline{p_j} + \Delta_{inter_j} + \Delta_{intra_j} \quad (4)$$

where  $\overline{p_j}$  is the nominal value of the process parameter,  $\Delta_{inter_j}$  is the inter-die variation of the process parameter and  $\Delta_{intra_j}$  is the intra-die variation of the process parameter. The inter-die variation is the same for all components (nets, gates) on a die. The intra-die process parameters vary within a die.

##### B.1. Intra-die Process Variations

Intra-die process variations are composed of two components: deterministic variation and random Gaussian variation. A gradient model accounts for deterministic

variations in intra-die process parameters while a random Gaussian noise model accounts for intra-die variations that are spatially correlated. The intra-die variations are represented as follows:

$$\Delta_{intra,i,j} = x_i A_j + y_i B_j + N(0, C_j) \quad (5)$$

where  $\Delta_{intra,i,j}$  is the intra-die variation in grid cell  $i$  (the grid model is described in the next section),  $x_i$  and  $y_i$  are the  $x$  and  $y$  coordinates of grid cell  $i$ ,  $A_j$  and  $B_j$  are the parameters of the gradient plane for process parameter  $j$  and  $N(0, C_j)$  is a multi-variate normal variable with mean 0 and covariance matrix  $C_j$  [3].

## B.2. Gradient Model

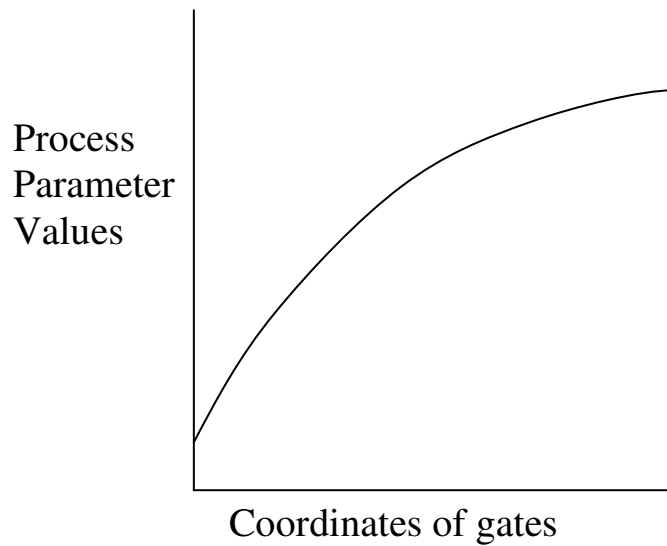


Figure 7. Gradient example.

Figure 7 shows an example of a process gradient across a die. It can be seen that as the coordinate location of the gates/interconnect increases, the amount of process parameter variation on the gates/interconnect also increases correspondingly, following a gradient. The gradient can be approximated by a plane equation  $(x_i A_j + y_i B_j)$  and is

included in our timing analysis. The constant in the standard plane equation is not included here, since it is included in  $\overline{p_j}$  in Equation 4.

### B.3. Spatial Correlations

To incorporate spatial correlation into our analysis, the die is partitioned into  $n \times n$  grid cells. A sample  $2 \times 2$  partition of a die is illustrated in Figure 8. The components within a grid cell have perfect correlation, with the correlation between different grid cells falling with their separation distance [12]. The correlation falls to zero within a few hundred microns. In our model, the correlation falls with the distance between the centers of two grid cells. The function for determining the correlation factor can be as simple as  $1 / (2 \cdot \text{distance between grid cells})$ .

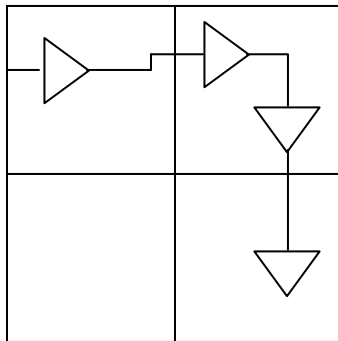


Figure 8.  $2 \times 2$  partition of a die.

A random variable is defined for every process parameter  $j$  in every grid cell. Hence, there are  $n^2 \cdot m$  random variables in our analysis, where  $m$  is the number of process parameters. Correlation exists only between random variables of the same process parameter.

#### B.4. Covariance Matrix

As explained in the previous section, spatial correlations are modeled by dividing the die into grids. The correlation between the grid cells is represented using a covariance matrix  $C$ . A covariance matrix  $C_j$  is defined for each process parameter  $j$ . For a  $n \times n$  partition of the die, the size of the covariance matrix is  $n^2 \times n^2$ . The value of each cell in the covariance matrix ( $cov(x, y)$ ) is the product of the correlation factor between grid cells  $x$  and  $y$ , represented by  $\rho_{x,y}$ , and the standard deviations of grid cells  $x$  and  $y$ , represented by  $\sigma_x$  and  $\sigma_y$  respectively, i.e.

$$cov(x, y) = \rho_{x,y} \cdot \sigma_x \cdot \sigma_y \quad (6)$$

The covariance matrix generation can be simplified greatly for large die sizes. The covariance matrix is symmetric since  $cov(x, y) = cov(y, x)$ . Hence it is sufficient to calculate only  $n^4/2$  values. Since the correlation falls to zero beyond  $k$  grid cells ( $k < n^2$ ), non-zero correlation factors exist only for grid cells that lie within a distance of  $k$  from the grid cell in consideration. Hence it is sufficient to calculate the correlation factors for grid cells that lie within an area of size  $2k \times 2k$  with the grid cell in consideration at the center. In other words, matrix  $C$  is band structured, i.e. each grid cell has non-zero correlation values with at most  $4 \cdot k^2$  grid cells and hence the covariance matrix  $C$  has at most  $4 \cdot k^2 \cdot n^2$  non-zero elements.

Sample covariance matrix values for grid '\*' with standard deviation of 1.0 for all grid cells and *correlation function* =  $1/(2 \cdot \text{distance between grid cells})$  are shown in Figure 9. Distance between two grid cells is defined as one plus the minimum number of grid cells between them.

0.25	0.25	0.25
0.5	0.5	0.25
*	0.5	0.25

Figure 9. A sample die with values in the covariance matrix for grid ‘\*’

#### B.5. Principal Component Analysis (PCA)

The  $n^2$  random variables for each process parameter  $j$  are correlated to each other with different amounts of correlation. For a given grid cell size, the number of random variables increases exponentially with the size of the die. Principal Component Analysis (PCA) can be used to make the analysis tractable [3, 7]. PCA transforms a set of correlated variables into a smaller set of principal components that are independent and orthonormal. The first principal component (i.e. the component with the highest eigenvalue) accounts for the maximum amount of variance represented by the  $n^2$  random variables. The second component accounts for the next largest fraction of the variance, and so on. In practice, a small number of principal components can be used to accurately model the variance [3, 7]. However, the maximum number of principal components for a grid of size  $n \times n$  is equal to the number of rows/columns in  $C_j$  which is equal to the number of individual grid cells, i.e.  $n^2$ .

PCA uses the covariance matrix  $C_j$  to transform the set of correlated random variables into a set of uncorrelated random variables with mean 0 and standard deviation

1. In this way, each correlated random variable (one for each grid cell) can be represented as:

$$V_{i,j} = \mu_{j,i} + a_{j,1} \cdot pc_{j,1} + a_{j,2} \cdot pc_{j,2} + \dots + a_{j,n^2} \cdot pc_{j,n^2} \quad (7)$$

where  $V_{i,j}$  is the original correlated random variable for grid  $i$  and process variable  $j$ ,  $\mu_{j,i}$  is the nominal value of parameter  $j$  in grid  $i$ ,  $a_{j,1}$  is the coefficient of principal component  $pc_{j,1}$  and so on until  $a_{j,n^2}$  and  $pc_{j,n^2}$ , and  $n^2$  is the total number of principal components. The principal components are the uncorrelated random variables with mean 0 and standard deviation 1. The coefficients  $a_{j,k}$  are calculated using the following formula:

$$a_{j,k} = \sqrt{\lambda_{j,l}} \cdot ev_{j,k,l} \cdot \sigma_{j,i} \quad (8)$$

where  $\lambda_{j,l}$  is the  $l^{th}$  eigenvalue of the covariance matrix  $C_j$  and  $ev_{j,k,l}$  is the  $k^{th}$  element of  $l^{th}$  eigenvector of  $C_j$  and  $\sigma_{j,i}$  is the standard deviation of process parameter  $j$  in grid  $i$ .

$V_{i,j}$  denotes the process parameter values. Although we have expressed the correlated random variable in terms of all  $n^2$  principal components, in reality it is sufficient to consider only the first few components. A circuit designer may decide on the number of principal components to be used in SSTA, depending on the time and accuracy tradeoff.

The delay contributed by the process variable  $j$  in the grid cell  $i$  can be obtained by multiplying  $V_{i,j}$  by the sensitivity of delay to this process parameter in this grid cell in accordance with our linear delay model. Hence the gate delay corresponding to this process variable  $j$  can be expressed as a linear function of the principal components. The

gate delay considering all the process parameters can be expressed as a linear function of all such principal components, i.e.

$$d = d_0 + a_1 \cdot pc_1 + a_2 \cdot pc_2 + \dots + a_{num} \cdot pc_{num} \quad (9)$$

where  $d_0$  is the nominal delay,  $a_1$  is the coefficient of principal component  $pc_1$  and so on until  $a_{num}$  and  $pc_{num}$  and  $num$  is the total number of useful principal components considering all process parameters. The coefficients of the principal components in Equation 9 (and in all following equations) include delay sensitivity as explained in the previous paragraph. The  $num$  value is decided by the circuit designer depending on the time and accuracy tradeoff. The variance of  $d$  can be calculated as the sum of the squares of the coefficients, i.e.

$$\sigma_d^2 = \sum_v^{num} a_v^2 \quad (10)$$

#### B.6. Covariance Between Paths

Covariance between paths 1 and 2 ( $cov(1,2)$ ) can be calculated as follows:

$$d_1 = d_{1,0} + a_{1,1} \cdot pc_1 + a_{1,2} \cdot pc_2 + \dots + a_{1,num} \cdot pc_{num}$$

$$d_2 = d_{2,0} + a_{2,1} \cdot pc_1 + a_{2,2} \cdot pc_2 + \dots + a_{2,num} \cdot pc_{num}$$

$$cov(1,2) = \sum_v^{num} a_{1,v} \cdot a_{2,v} \quad (11)$$

#### C. Crosstalk

As described earlier, the presence of crosstalk makes the problem of calculating path delay complicated. Crosstalk is analyzed as follows: given the delay distribution due to process variations at two nodes (victim X and aggressor Y) that have opposite



transitions, it is possible to calculate the distribution of Y subtracted from X (denoted X-Y) with mean  $\mu_{x-y}$  and variance  $\sigma_{x-y}^2$  as follows:

$$\mu_{x-y} = \mu_x - \mu_y \quad (12)$$

$$\sigma_{x-y}^2 = \sigma_x^2 + \sigma_y^2 - 2 \cdot \sigma_x \cdot \sigma_y \cdot \rho_{x,y} \quad (13)$$

where  $\mu_x$ ,  $\sigma_x$  and,  $\mu_y$ ,  $\sigma_y$  are the means and standard deviations of X and Y respectively and  $\rho_{x,y}$  is the correlation factor between X and Y.

The following initial assumptions are made in the linear crosstalk model: the victim and the aggressor signals have the same slew rate and similar driver strength; hence the worst case delay degradation occurs when the signals transition at the same time. Sometimes, the worst case delay increase may occur with different slew rates or different driver strengths as well [17]. In such cases, our crosstalk model will not estimate the delay correctly. But our assumption of the same slew rates greatly simplifies the analysis.

When the delay difference between X and Y is 0 ( $|X - Y| = 0$ ), the transitions are perfectly aligned and there is maximum delay increase on the victim path. As the delay difference increases ( $|X - Y| > 0$ ), the extra victim line delay falls to zero, as shown in Figure 10 (the delay values in the figure are for explanation purposes only) [26]. The relationship between increase in (victim) delay and delay difference between paths can be approximated as linear.

$$d_{inc} = d_{inc,max} - slope \cdot d_{diff} \quad (14)$$

where  $d_{inc}$  is the victim delay increase due to crosstalk,  $d_{diff}$  is the delay difference between the victim and the aggressor,  $d_{inc,max}$  is the maximum victim delay increase due to crosstalk,  $slope$  is the slope of the line and is equal to  $d_{inc,max} / d_{diff,max}$  and  $d_{diff,max}$  is the difference between victim and aggressor signal transition times beyond which there is no delay increase on the victim.

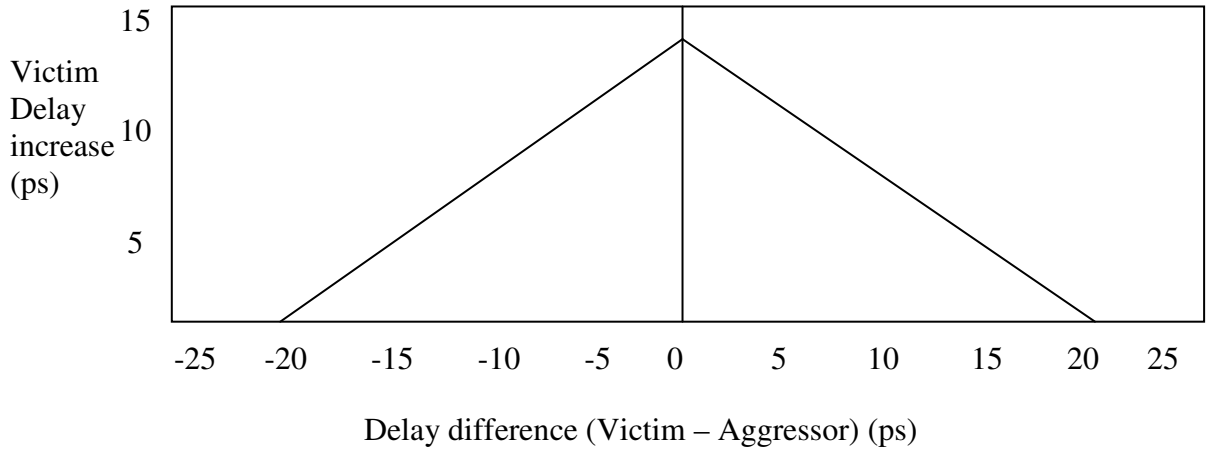


Figure 10. Crosstalk delay increase curve.

Using the X-Y distribution and the crosstalk delay increase Equation 14, a normal distribution of delay increase is approximated using piecewise linear (PWL) analysis [2]. In PWL analysis, the distribution is divided into a number of segments as in Figure 11 with the delay difference assumed constant within each segment. The delay increase on the victim is calculated for each segment based on the delay difference. The mean ( $\mu_{Xtalk}$ ) and standard deviation ( $\sigma_{Xtalk}$ ) of the victim delay increase normal distribution is calculated using the following formulae:

$$\mu_{Xtalk} = \sum_r^{seg} d_r p_r \quad (15)$$

$$\sigma_{x_{talk}} = \sqrt{\frac{1}{seg} \sum_r^{seg} (d_r p_r - \mu_{x_{talk}})^2} \quad (16)$$

where  $d_r$  and  $p_r$  are the victim delay increase and probability of segment  $r$  respectively and  $seg$  is the number of segments.

Since we are using linear delay equations, the mean of the victim delay increase is added to the mean path delay while the standard deviation of victim delay increase becomes an additional principal component to the path delay Equation 9 to obtain Equation 17. The total number of principal components is equal to  $num + 1$ .

$$d = d_0 + a_1 \cdot pc_1 + a_2 \cdot pc_2 + \dots + a_{num} \cdot pc_{num} + a_{x_{talk}} \cdot pc_{x_{talk}} \quad (17)$$

Equation (17) is rewritten as follows:

$$d = d_0 + a_1 \cdot pc_1 + a_2 \cdot pc_2 + \dots + a_{total-1} \cdot pc_{total-1} + a_{total} \cdot pc_{total} \quad (18)$$

where  $total = num + 1$ .

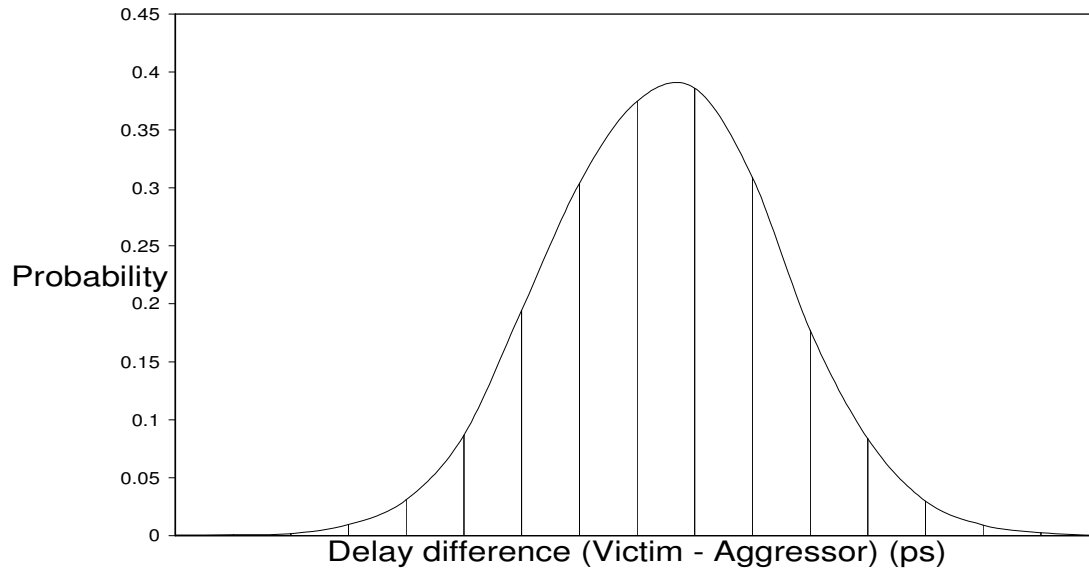


Figure 11. Delay difference curve for piece-wise linear analysis.

Our crosstalk analysis makes another set of approximations. First, the delay increase due to crosstalk is assumed to be independent (or orthogonal) with other principal components (process variations). Correlation between the paths is still considered in the analysis. Second, all the aggressors are considered simultaneously. For example, if there are two aggressor nets, the effect of the first aggressor and the second aggressor are computed separately but incorporated simultaneously into the victim's delay before the next gate on the path is analyzed. Helpers are not considered in this crosstalk analysis.

Correlation factor between crosstalk and process variations can alternate between positive and negative values without any regularity due to the non-monotonous nature of crosstalk and its dependence on the instantaneous delay of the victim and multiple neighboring switching nets. Approximating delay increase due to crosstalk as independent of process is reasonable, since the change in delay due to crosstalk is normally much less than due to process variation, and it drastically simplifies the analysis.

Helpers can decrease the delay on the victim nets and thereby possibly prevent a setup time violation, while at the same time; it could lead to a hold time violation since the data may arrive too early at a latch/flip-flop. Latches are prone to hold time violations unlike flip-flops since data signals can pass through a latch as long as the clock signal is high and thus arrive too early at the next latch. Considering all aggressors and no helpers is conservative when timing flip-flop based designs, but must be modified to consider short paths in latch-based designs.

Rather than considering aggressors simultaneously, other approaches would be to consider them one by one or in decreasing order of coupling capacitance or delay impact. The effect of crosstalk on the aggressor paths are not considered in our analysis although the effect of process variations is taken into consideration. Including the effect of crosstalk on the aggressor paths may lead to possible ‘chicken and egg’ problems and computational inefficiency. It is to be noted that searching for a worst case crosstalk delay increase on a gate in the path may not necessarily lead to the worst case path delay. These approximations are still much less conservative than a worst-case corner-based approach.

#### *D. Statistical Timing Analysis*

In a path based approach to statistical static timing analysis, each path is analyzed separately to compute their path delay distribution. The gates on the path are added one by one to the path delay through an addition operation. Once the delay distributions of all paths are computed, the maximum of all the path delay distributions is calculated through a maximum operation.

##### **D.1. SUM and MAX Operations**

In statistical static timing analysis (SSTA), it is sufficient to obtain the mean and standard deviation of the delay. As described earlier, the standard deviation can be easily calculated from the coefficients of the principal components using Equation 10. A SUM operation is required to add a gate delay to the existing path as we move along the path from the primary input towards the primary output. A MAX operation of all the longest paths is needed to calculate the mean and the standard deviation of the critical length of

the circuit. Both these operations can be performed efficiently with the coefficients of principal components [3, 31, 32] and are explained below.

## D.2. SUM Operation

The SUM operation at each gate on a path ( $d_{sum} = d_1 + d_2$ ) can be computed as follows:

$$\begin{aligned}
 d_1 &= d_{1,0} + a_{1,1} \cdot pc_1 + a_{1,2} \cdot pc_2 + \dots + a_{1,total} \cdot pc_{total} \\
 d_2 &= d_{2,0} + a_{2,1} \cdot pc_1 + a_{2,2} \cdot pc_2 + \dots + a_{2,total} \cdot pc_{total} \\
 d_{sum} &= d_{sum,0} + a_{sum,1} \cdot pc_1 + a_{sum,2} \cdot pc_2 + \dots + a_{sum,total} \cdot pc_{total}
 \end{aligned} \tag{19}$$

where  $d_{sum,0} = d_{1,0} + d_{2,0}$ ,  $a_{sum,1} = a_{1,1} + a_{2,1}$  and so on until  $a_{sum,total}$ . The standard deviation of  $d_{sum}$  can be calculated using Equation 10 on the new set of coefficients.

## D.3. MAX of Longest Paths

Upon evaluating the distribution of each longest path individually using SUM operations, the maximum is calculated of the distribution of the longest paths. A closed-form formulae to calculate the maximum of two normal distributions is available in [31]. The maximum distribution of  $n$  longest paths in our approach is calculated by repeatedly applying the MAX function to two normal distributions. The paths are sorted by ‘nominal delay + standard deviation’ ( $\mu + \sigma$ ) before the maximum distribution is computed, starting with the longest path. Figure 12 illustrates the order in which the maximum of all the longest paths is calculated. A statistical timing analysis approach that has  $num\_longest$  number of longest paths, requires the MAX function to be used  $O(num\_longest)$  times.

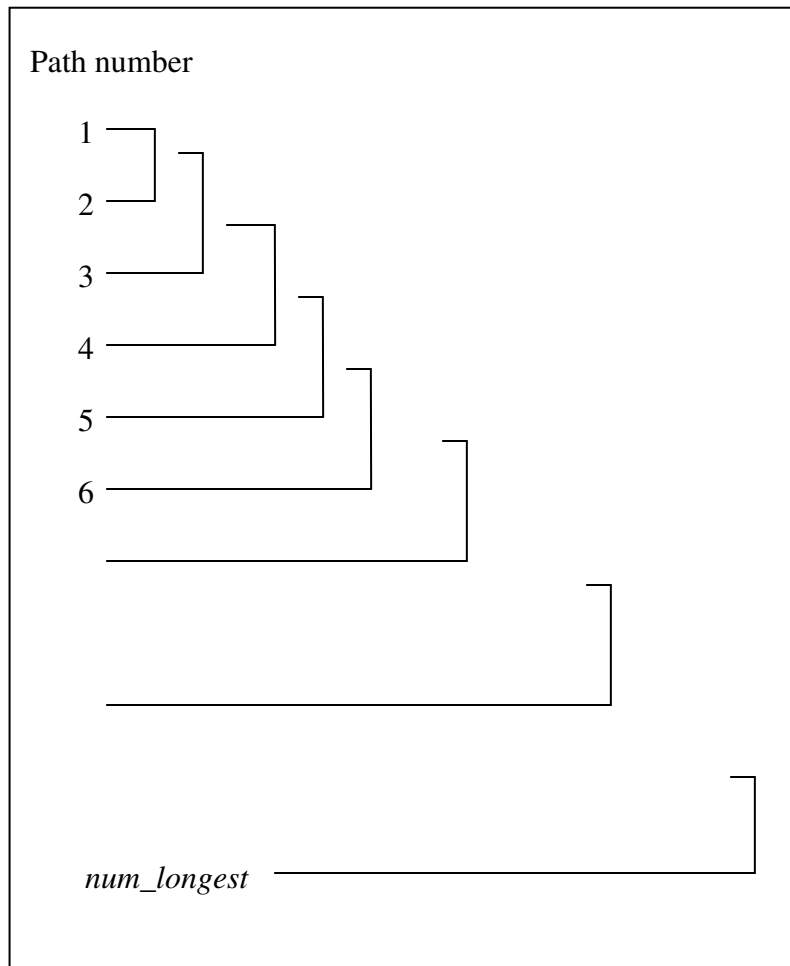


Figure 12. Order of maximum distribution computation.

This method of repeatedly applying the MAX function may introduce errors in the final maximum distribution. An alternative approach would be to do pair-wise calculations of the maximum distribution of two paths with similar values of  $\mu + \sigma$ , in a tree-like fashion. In a tree approach to compute the maximum of all the longest paths, the MAX function is used  $O(\log num\_longest)$  times. Hence, a tree approach is faster and could potentially be more accurate.

The maximum distribution ( $\mu_{d_{\max}}, \sigma_{d_{\max}}$ ) of two normal distributions with means ( $\mu_x, \mu_y$ ) and standard deviations ( $\sigma_x, \sigma_y$ ) and a correlation factor of ( $\rho_{x,y}$ ) between the distributions is calculated as follows:

The maximum distribution takes the form:

$$d_{\max} = \mu_{d_{\max}} + a_1 \cdot pc_1 + a_2 \cdot pc_2 + \dots + a_{total} \cdot pc_{total} \quad (20)$$

where  $a_1, a_2, \dots, a_{total}$  are the coefficients of principal components  $pc_1, pc_2, \dots, pc_{total}$  respectively.

Case 1: Standard deviations are equal ( $\sigma_x = \sigma_y$ ) and correlation factor is 0 ( $\rho_{x,y} = 0$ ),

$$d_{\max} = \begin{cases} d_1, & \text{if } \mu_1 \geq \mu_2 \\ d_2, & \text{otherwise} \end{cases} \quad (21)$$

Case 2: Standard deviations are not equal ( $\sigma_x \neq \sigma_y$ ) or correlation factor is not equal to 0 ( $\rho_{x,y} \neq 0$ ),

We define two constants ( $\alpha$  and  $\beta$ ) as follows:

$$\alpha = \sqrt{\sigma_x^2 + \sigma_y^2 - 2 \cdot \sigma_x \cdot \sigma_y \cdot \rho_{x,y}}$$

$$\beta = (\mu_x - \mu_y) / \alpha$$

We define two functions  $\varphi(x)$  and  $\phi(x)$  as follows:

$$\varphi(x) = \frac{1}{\sqrt{2\Pi}} \exp(-x^2 / 2)$$

$$\phi(x) = \frac{1}{\sqrt{2\Pi}} \int_{-\infty}^x \exp(-y^2 / 2) \cdot dy$$



According to [33],

$$\phi(x) = \frac{1}{2}[1 + \operatorname{erf}(x/\sqrt{2})]$$

$$\phi(-x) = \frac{1}{2}[1 - \operatorname{erf}(x/\sqrt{2})]$$

The program to calculate the error function ( $\operatorname{erf}(x)$ ) is available at [34]. The first moment ( $d'_{\max}$  or  $E(d_{\max})$ ) and the second moment ( $d''_{\max}$  or  $E(d_{\max}^2)$ ) of the max distribution are calculated as follows:

$$E(d_{\max}) = \mu_x \cdot \phi(\beta) + \mu_y \cdot \phi(-\beta) + \alpha \cdot \phi(\beta)$$

$$E(d_{\max}^2) = (\mu_x + \sigma_x^2) \cdot \phi(\beta) + (\mu_y + \sigma_y^2) \cdot \phi(-\beta) + (\mu_x + \mu_y) \cdot \alpha \cdot \phi(\beta)$$

We know that the mean and the standard deviation of the distribution can be calculated using the first and second moments as follows:

$$\mu_{d_{\max}} = E(d_{\max}) \quad (22)$$

$$\sigma_{d_{\max}}^2 = E(d_{\max}^2) - E(d_{\max})^2$$

$$\sigma_{d_{\max}}^2 = (\mu_x + \sigma_x^2) \cdot \phi(\beta) + (\mu_y + \sigma_y^2) \cdot \phi(-\beta) + (\mu_x + \mu_y) \cdot \alpha \cdot \phi(\beta) - \mu_{d_{\max}}^2 \quad (23)$$

The coefficients of the principal components of the new normal distribution are calculated as follows:

$$a_r = \operatorname{cov}(d_{\max}, pc_r)$$

$$a_r = \frac{\sigma_x \cdot k_{xr} \cdot \phi(\beta) + \sigma_y \cdot k_{yr} \cdot \phi(-\beta)}{\sigma_{d_{\max}}} \quad (24)$$

But since there is a potential for mismatch between the standard deviation calculated using the coefficients ( $\sigma_d = \sqrt{\sum_v^{total} a_v^2}$ ) and the standard deviation calculated

using the closed-form formulae in Equation 23, the coefficients ( $a_r$ ) are normalized to reduce the standard deviation mismatch and potential errors in further calculations using the coefficients and standard deviation:

$$s_0 = \sqrt{\sum_v^{total} a_v^2} \quad (25)$$

$$a_v = a_v \cdot \frac{\sigma_{d_{max}}}{s_0} \quad (26)$$

#### D.4. Longest Path Generation

One of the major challenges of using a path-based timing analysis approach is the complexity involved in selecting the set of longest paths. We make use of CodGen [11] to efficiently generate a set of globally longest paths that could be the largest on some chip. CodGen is primarily used to generate the K Longest Paths through each Gate (KLPG) in the circuit, using robust or non-robust sensitizability analysis, producing the input patterns to test the path. This tool uses direct implications [35], forward trimming [36], smart-PERT [11] and final justification [37] algorithms to trim the search space.

CodGen was modified to generate the globally longest paths, with most sensitizability checks turned off to speed up the path generation. The sensitizability checks that were turned off are forward trimming, Smart-PERT and final justification. Enough checks were left in place so that most generated paths are sensitizable. In this work, CodGen uses only direct implications to eliminate the false paths. This is a significant improvement in accuracy over approaches that only consider structurally longest paths.

#### D.5. Aggressor Path Generation

Crosstalk analysis requires the generation of opposite transitions on aggressor nets within the switching window of the victim net. As explained earlier, the relative increase in delay due to crosstalk is smaller than the effect of process variations. CodGen has been modified to generate a list of paths to each potential aggressor net that have the required transition within the victim's switching window. Nominal delay of the gates is used during aggressor path generation. These side paths use the same sensitizability checks as the target path. In many cases, no side path can be found, so the aggressor net is either stable or switches in the same direction as the victim during the victim's switching window. Such aggressor nets are ignored (assumed to have stable values) in timing analysis of the corresponding longest path. CodGen generates paths in descending order of the nominal delay. Hence while generating the aggressor paths, the first path that is within the switching window will be selected for each aggressor net. This process is repeated for every victim-aggressor pair. It is to be noted that we are not always selecting the aggressor paths that are closest to the victim delay. An attempt to generate the aggressor path that is closest to the victim delay may be time consuming as we are aiming to find the "best" path.

## IV. IMPLEMENTATION AND RESULTS

### A. Implementation Details

SSTA has been performed on ISCAS85 circuits [38] designed in a TSMC 180 nm, 4-metal layer technology. Cadence Silicon Ensemble<sup>TM</sup> was used for circuit layout generation while parasitics were extracted by a 2.5D extractor Cadence HyperExtractor. The SSTA has been implemented in 5000 lines of code in C++ using Visual Studio and experiments run on a Windows XP machine with a 930 MHz Pentium 3 processor and 256 MB of memory. Transistor gate length, metal width, metal thickness and interlayer dielectric (ILD) thickness are the process variables considered in this analysis. The amount of variation in these process variables are shown in Table 1 [39]. The standard deviations for metal width, metal thickness and ILD thickness are the same for all four metal layers. The amount of process variation is divided equally between inter-die variations and intra-die variations, as in [3]. Gradients account for 20% of intra-die variations while the remaining 80% is accounted for by spatial correlated random variation.

**Table 1. Standard deviation of process variables.**

<b>Process variable</b>	<b>Standard Deviation</b>
Gate length	3.3%
Metal width	10%
Metal thickness	16.7%
ILD Thickness	16.7%

Monte Carlo (MC) simulations have been performed to verify the results of SSTA with crosstalk ( $SSTA_{\text{xtalk}}$ ). 100,000 iterations were performed for each circuit in MC analysis so that the MC sample variation is small. The default individual grid cell size is  $150\ \mu\text{m}$  by  $150\ \mu\text{m}$  for all circuits and the default correlation distance is  $450\ \mu\text{m}$  (i.e. there is no correlation beyond  $450\ \mu\text{m}$ ). The grid dimensions ( $n \times n$ ) for each circuit depends on the die area, and are provided in Table 2 for the default individual grid cell size of  $150\ \mu\text{m}$  by  $150\ \mu\text{m}$ . CodGen uses the nominal delay of the gates for globally longest and aggressor paths generation. CodGen is used to generate the 200 globally longest paths, and the aggressors for each net on the longest paths that switch within a range of  $\pm 30\%$  around the nominal delay of the victim. The MC and SSTA analyses both use the same delay model, so the comparisons do not consider delay model error. Previous research shows that the linear delay model introduces only a small error [29]. The linear crosstalk model is validated using circuit simulation and the results are presented in Section IV.F. The time for PCA (performed with MATLAB on a Sun SPARC V9 processor with Solaris 8.0 operating system and 8 GB of RAM) is less than 5 s for  $10 \times 10$  grids, and negligible for the circuits analyzed here. The overall flow of the SSTA tool is outlined in Figure 13.

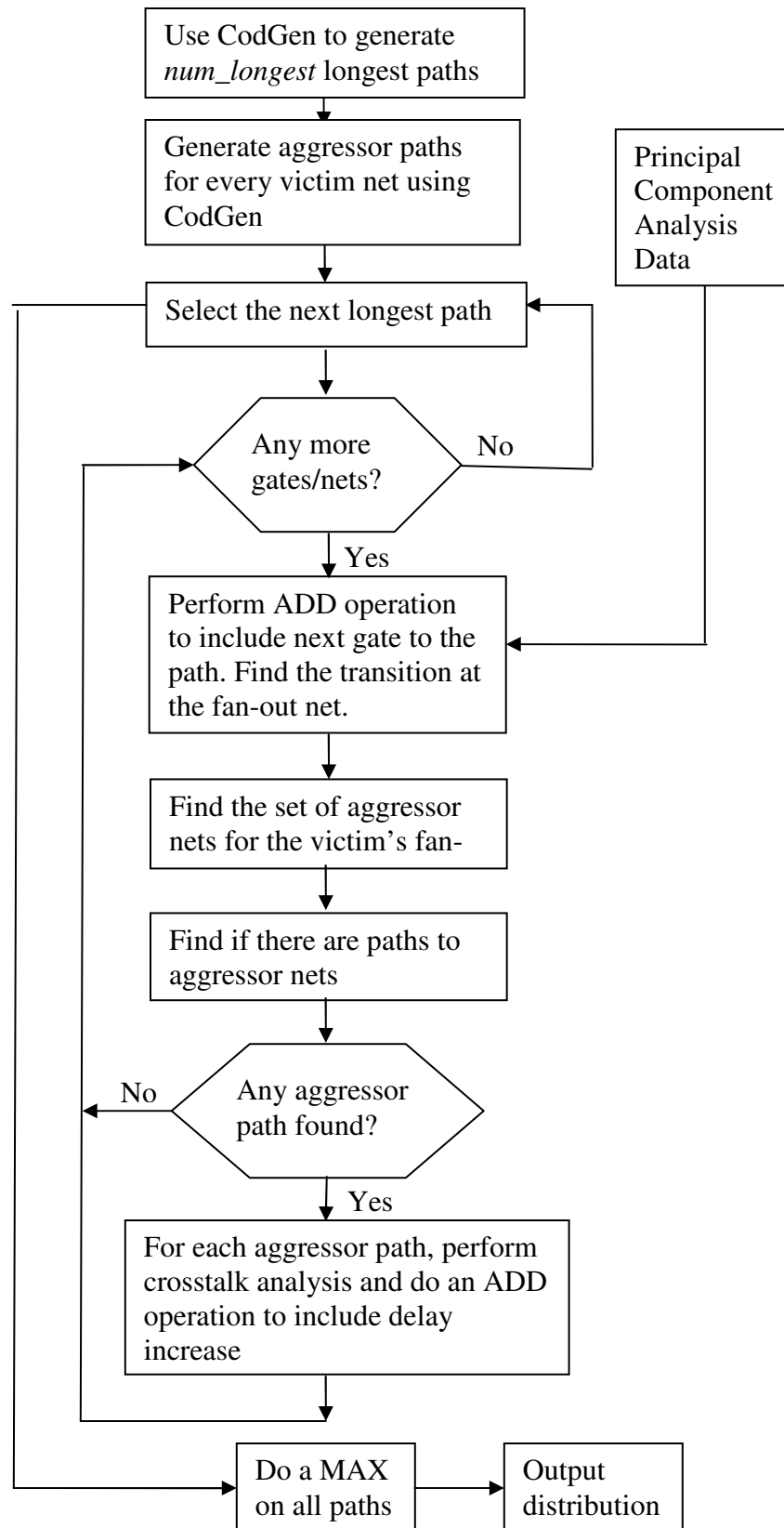


Figure 13. SSTA tool flow.

B. Accuracy of SSTA Model

**Table 2. Analysis of SSTA<sub>xtalk</sub> and Monte Carlo method.**

Circuit	Grid dimension	Monte Carlo (ns)		SSTA <sub>xtalk</sub> (ns)		$\frac{(SSTA_{xtalk} - MC)}{MC}$ (%)	
		Mean	SD	Mean	SD	Mean	SD
C432	1×1	0.43	0.02	0.43	0.02	0.04	-1.67
C499	2×2	0.65	0.03	0.65	0.03	0.02	1.43
C880	2×2	0.77	0.06	0.77	0.06	0.17	3.25
C1355	2×2	0.81	0.09	0.82	0.08	0.70	-13.14
C1908	2×2	0.82	0.06	0.83	0.05	0.81	-8.62
C2670	2×2	1.14	0.07	1.14	0.07	0.51	1.65
C3540	3×3	1.34	0.07	1.34	0.07	0.11	-0.47
C5315	3×3	1.09	0.08	1.09	0.08	-0.05	1.34
C6288	3×3	3.06	0.67	3.06	0.68	0.00	1.34
C7552	4×4	0.97	0.06	0.97	0.05	0.38	-15.04

The mean and standard deviation (SD) of MC and SSTA<sub>xtalk</sub> are provided in Table 2 along with the percentage error due to our approach. In comparison with Monte Carlo simulation, the average SSTA error is only 0.27% in mean and -2.99% in standard deviation. The paths were sorted by decreasing order of ‘nominal delay + standard deviation’ before the maximum distribution is computed. It is found that computing the maximum distribution in the decreasing order of the paths’ mean delay reduces the error in standard deviation for C1908 and C7552 from -8.62% and -15.04% to 0.88% and



1.89% respectively. The initial delay distribution of the top 200 longest paths from CodGen for C1355 is very narrow, i.e. the delay difference between the longest path and the 200<sup>th</sup> longest path is smaller than that of any other circuit. The narrow range could have possibly led to an accumulation of errors in standard deviation during the computation of the maximum distribution. The normalization of standard deviation to reduce the standard deviation mismatch in Equation 26 may not have been effective for this narrow range. A tree approach to compute the maximum of all longest paths may reduce the error for C1355. In Figure 14, we plot the PDF of C6288 for  $SSTA_{xtalk}$  and MC analysis. It can be seen that the distributions match each other closely.

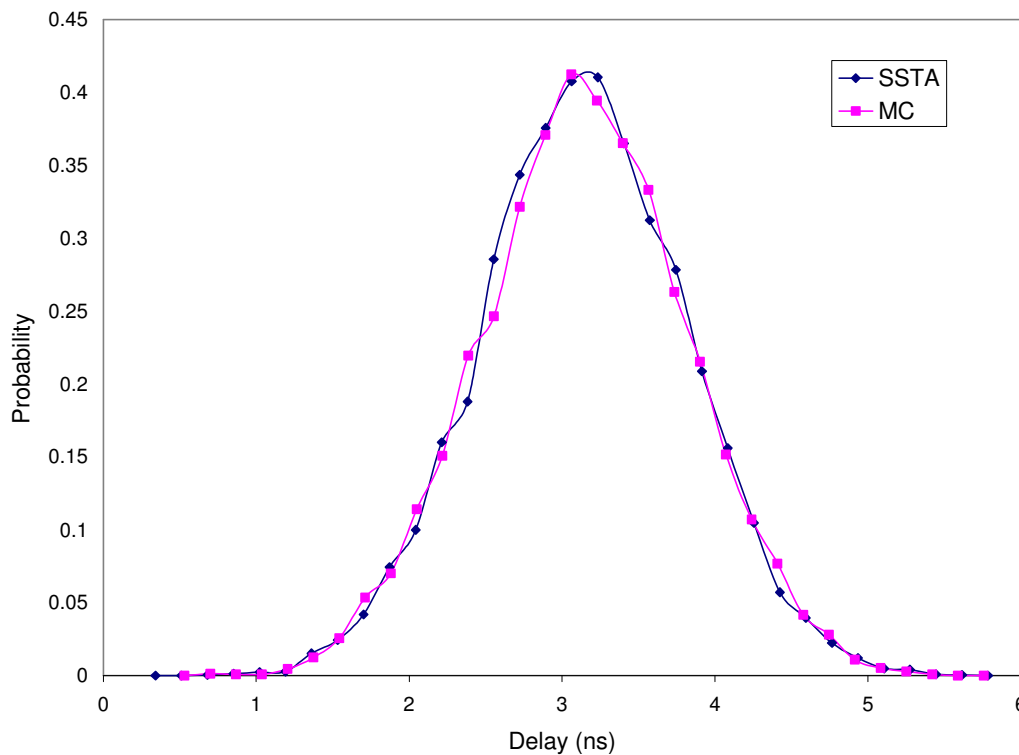


Figure 14. Comparison of PDF plots of MC and  $SSTA_{xtalk}$  for circuit C6288.

**Table 3. Simulation time for SSTAxTalk analysis.**

Circuit	Simulation time (MM:SS)			
	Longest paths (CodGen)	Aggressor paths (CodGen)	SSTA <sub>xtalk</sub>	Total
C432	00:01	00:17	00:02	00:20
C499	00:01	00:13	00:01	00:15
C880	00:02	00:15	00:06	00:23
C1355	00:01	05:58	00:12	06:11
C1908	09:44	00:56	00:09	10:49
C2670	00:12	02:18	00:14	02:44
C3540	01:54	10:05	00:13	12:12
C5315	00:19	01:39	00:05	02:03
C6288	17:15	28:51	00:59	47:05
C7552	00:13	01:43	00:12	02:08

Table 3 lists the execution times for longest path generation, aggressor path generation and actual SSTA<sub>xtalk</sub> analysis. The large time taken to generate the longest paths for C1908 and C6288 is due to the large number of false paths in these circuits. This could be reduced at the expense of accuracy by turning off false path checks. A large number of potential aggressors for each victim are considered while generating aggressor paths. For example, C6288 has about 120 gates on the longest paths and each victim net has on average 3 potential aggressors. As a result, aggressor path generation is costly. The time for performing the actual timing analysis SSTA<sub>xtalk</sub> ranges from 1 s to 59

s, averaging 12.4 s. For a given circuit, this means that different process variation analyses can be quickly run, since the path generation need only be done once per circuit.

### C. Importance of Crosstalk in SSTA

**Table 4. Importance of considering crosstalk in SSTA.**

Circuit	SSTA <sub>no-xtalk</sub> (ns)		$\frac{(SSTA_{xtalk} - SSTA_{no-xtalk})}{SSTA_{no-xtalk}}$ (%)	
	Mean	SD	Mean	SD
C432	0.40	0.01	7.72	13.88
C499	0.55	0.03	17.46	-3.60
C880	0.72	0.06	7.10	3.48
C1355	0.74	0.07	10.43	14.64
C1908	0.77	0.04	7.14	25.06
C2670	1.06	0.06	8.20	13.37
C3540	1.27	0.06	5.89	15.16
C5315	1.03	0.07	6.53	2.96
C6288	2.97	0.67	3.16	1.21
C7552	0.90	0.03	8.39	44.30

Timing analysis without crosstalk (SSTA<sub>no-xtalk</sub>) has been performed on circuits to show the importance of considering crosstalk in SSTA. The mean and standard deviation of SSTA<sub>no-xtalk</sub> for the ISCAS 85 circuits are given in Table 4, along with the fraction of the mean and standard deviation that is due to crosstalk.

It can be seen that the average mean delay increase due to crosstalk is 8.2%. Hence, an analysis without crosstalk can significantly underestimate circuit delay. Circuit simulation on non-robust longest paths (generated by CodGen and used in our SSTA analyzes) was carried out using Cadence Spectre tool on ISCAS85 circuits (C432, C880) to validate these results. Only the first 200 vectors were simulated for each circuit since the circuit simulation takes a very long time. It was found that the mean delay increase due to crosstalk was 10.3% and 10.7% for C432 and C880 respectively, which is close to the results of our SSTA model.

The standard deviation for all the circuits (except C499) increases since the effect of crosstalk is different on different paths. An analysis of the longest paths in C499 reveals that mean and standard deviation increases in  $SSTA_{\text{xtalk}}$ . But, the amount of increase in standard deviation is very small relative to its standard deviation in  $SSTA_{\text{no-xtalk}}$ . Also, correlation between longest paths decreases in  $SSTA_{\text{xtalk}}$  over  $SSTA_{\text{no-xtalk}}$  for C499 (this happens for other circuits as well). These two factors force the standard deviation of  $SSTA_{\text{xtalk}}$  to be smaller than that of  $SSTA_{\text{no-xtalk}}$ . Figure 15 shows the PDF plots for circuit C7552 with and without crosstalk.

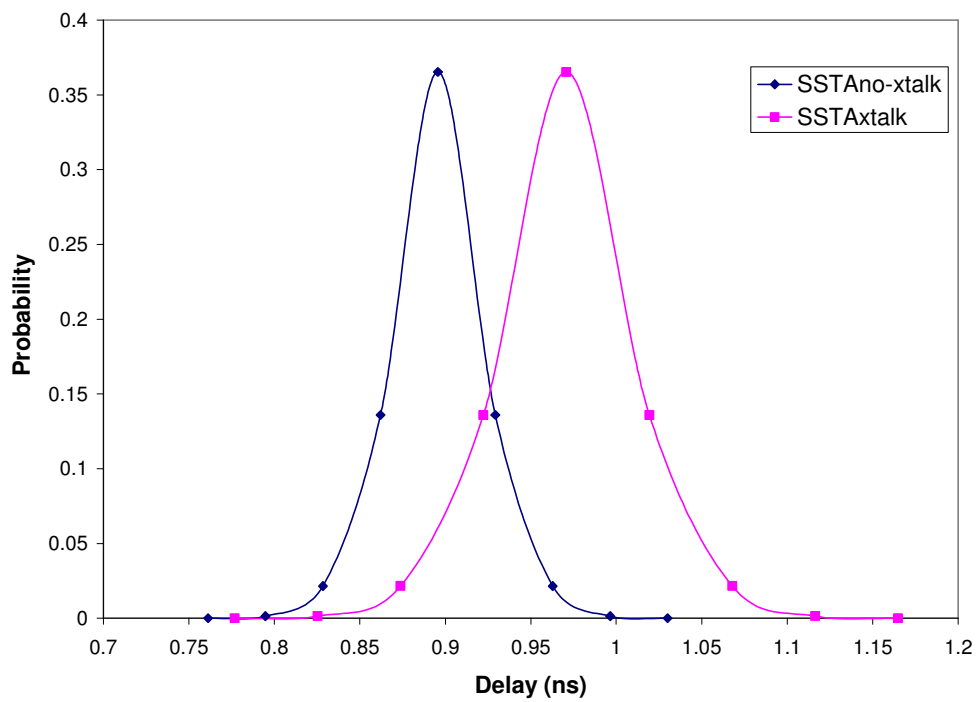


Figure 15. Importance of considering crosstalk in SSTA for C7552.

*D. Effect of Different Grid Sizes on Delay Distribution*

**Table 5. Comparison of standard deviation of SSTA results for various grid sizes**

Circuit	Individual grid sizes							
	150 $\mu\text{m}$	75 $\mu\text{m}$	50 $\mu\text{m}$	37.5 $\mu\text{m}$	30 $\mu\text{m}$	25 $\mu\text{m}$	18.75 $\mu\text{m}$	15 $\mu\text{m}$
C432	0.01416	0.01375	0.01397	0.01384	0.01405	0.01395	0.01399	0.01403
C499	0.03003	0.02997	0.03020	0.03006	0.03053	0.03027	0.03045	0.03052
C880	0.06049	0.06037	0.06133	0.06103	0.06193	0.06137	0.06179	0.06184
C1355	0.06847	0.06953	0.07015	0.06993	0.07083	0.07052	0.07065	0.07087
C1908	0.04302	0.04373	0.04465	0.04416	0.04478	0.04462	0.04469	0.04484
C2670	0.05848	0.05897	0.05973	0.05926	0.05971	0.05960	0.05972	0.05971
C3540	0.06165	0.06116	0.06159	0.06113	0.06151	0.06144	0.06156	0.06150
C5315	0.07435	0.07439	0.07478	0.07420	0.07451	0.07462	0.07454	0.07453
C6288	0.67256	0.67198	0.67912	0.67425	0.67832	0.67783	0.67857	0.67886
C7552	0.03360	0.03333	0.03337	0.03328	0.03342	0.03338	0.03342	0.03346

The default size of each individual grid in our analyses is 150  $\mu\text{m}$ . This number was chosen based on the correlation distance and the maximum die area of the ISCAS 85 circuits, in an attempt to balance accuracy and computation effort. But this number is not fixed. The grid size is specified by the circuit designer depending on the accuracy vs. time trade-off. Modeling each logic cell with an individual grid cell will lead to the most accurate results in the analysis, but the number of grid cells increases exponentially with the die area and hence the time for SSTA also increases. As the individual grid cell size is decreased, the delay distribution is expected to converge towards the exact value since the grid is just a stepwise approximation of the spatial correlation function.

The correlation structure affects only the standard deviation of the distribution. The nominal delay of each longest path remains the same for analyses with varying grid sizes. Due to the change in path correlation, the mean of the maximum distribution varies by a small amount as the grid size changes. SSTA analyses (without crosstalk) have been performed on seven other grid sizes: 75  $\mu\text{m}$ , 50  $\mu\text{m}$ , 37.5  $\mu\text{m}$ , 30  $\mu\text{m}$ , 25  $\mu\text{m}$ , 20  $\mu\text{m}$ , and 15  $\mu\text{m}$ . The standard deviation for these analyses are presented in Table 5. Figure 16 shows the standard deviation of circuit C1355 for various grid sizes. The same maximum correlation distance (of 450  $\mu\text{m}$ ) and correlation function has been maintained for the analyses.

As expected, the standard deviation converges as the grid size is reduced. Since the correlation between gates in two grid cells is measured as a function of the distance between the center points of the individual grids, the correlation factor changes as the grid cell size is reduced even though the distance between any two gates or interconnects remain fixed. Hence, the standard deviation does not converge to an exact value, but varies within a small range. The standard deviation does not increase or decrease monotonically because it depends on the spread of the longest paths over the die area, i.e. the correlation between paths is different for various grid sizes.

Reducing the grid cell size while keeping the same correlation distance decreases the area of the fully correlated region. Reducing the grid cell size and the correlation distance at the same time increases the independence of the process variations between gates in any two grids.

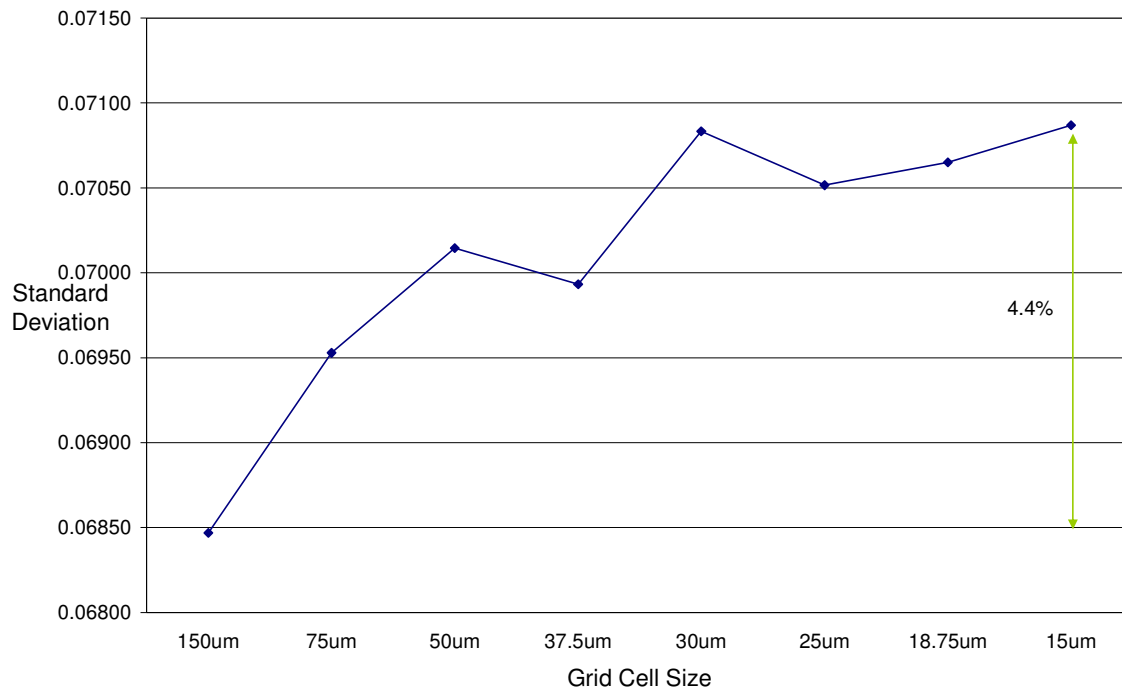


Figure 16. Effect of varying grid sizes for C1355.

### E. Correlation vs Independence

SSTA analyses have been performed with different correlation structures ranging from fully correlated to completely independent to understand the effect of correlation on the path distribution. Fully correlated process variables ( $SSTA_{full\_corr}$ ) have a chip delay mean that is smaller than that of the independent process variables ( $SSTA_{zero\_corr}$ ) while the standard deviation of the former ( $SSTA_{full\_corr}$ ) is larger than that of the latter ( $SSTA_{zero\_corr}$ ). This is valid because in a fully correlated structure (correlation factor = 1), the process variables are either increasing or decreasing simultaneously and hence the standard deviation is as large as possible. In an independent structure (correlation factor = 0), the process variables do not increase or decrease simultaneously, i.e. the joint distribution is always smaller than that of the fully correlated process variables. Although



the nominal delay of each longest path remains the same, the decrease in correlation between paths increases the mean of the maximum distribution.

**Table 6. Comparison of the impact of different correlation structures on SSTA with crosstalk.**

Circuit	SSTA <sub>full_corr</sub> (ns)		SSTA <sub>zero_corr</sub> (ns)		$\frac{(SSTA_{zero\_corr} - SSTA_{full\_corr})}{SSTA_{full\_corr}}$ (%)	
	Mean	SD	Mean	SD	Mean	SD
C432	0.43	0.02	0.43	0.02	0.00	0.00
C499	0.65	0.03	0.65	0.03	0.00	0.00
C880	0.77	0.06	0.77	0.06	-0.02	-0.71
C1355	0.83	0.09	0.82	0.08	2.12	6.90
C1908	0.84	0.06	0.83	0.06	1.25	9.22
C2670	1.15	0.07	1.14	0.07	0.23	-5.99
C3540	1.35	0.07	1.35	0.07	-0.03	-5.92
C5315	1.10	0.08	1.09	0.08	0.43	-8.96
C6288	3.08	0.63	3.06	0.75	0.58	-15.72
C7552	0.97	0.05	0.97	0.05	0.00	-2.68

Table 6 lists the mean and standard deviation for SSTA<sub>full\_corr</sub> and SSTA<sub>zero\_corr</sub> and the percentage increase in mean and standard deviation for SSTA<sub>zero\_corr</sub> against SSTA<sub>full\_corr</sub>. The average increase in mean is 0.45%, while the average decrease in standard deviation is 2.39%. The above analyses include crosstalk. Figure 17 shows the PDF of C6288 with different correlation structures: full correlation, partial correlation

and zero correlation. The partial correlation case corresponds to the default analysis of C6288 with crosstalk ( $SSTA_{\text{xtalk}}$ ), which has a correlation distance of  $450 \mu\text{m}$ . As can be seen, the mean is similar in all three cases while the standard deviation increases with correlation.

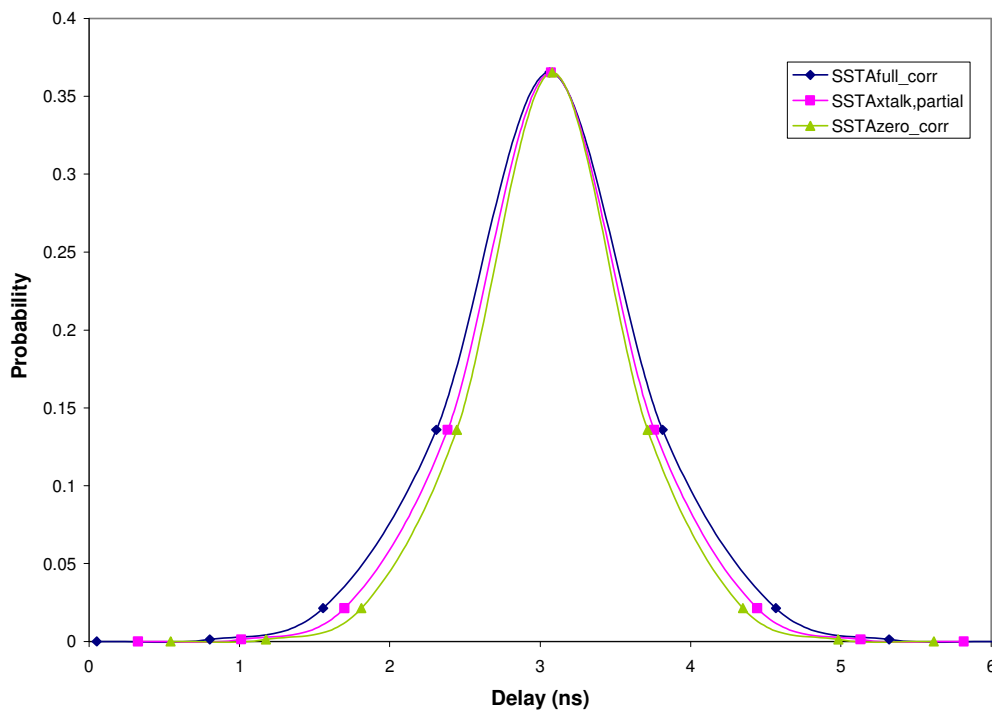


Figure 17. Comparison of SSTA results for different correlation structures for C6288.

The mean and standard deviation for C432 and C499 do not change as all the gates for these circuits are located in a single grid cell for the default individual grid cell size of  $150 \mu\text{m}$  by  $150 \mu\text{m}$ . Even though the die is divided into 2 by 2 grids for C499, all the gates/interconnects on the longest paths are located in only one grid cell, so it also acts as fully correlated.

The mean and standard deviation for C1355 and C1908 do not follow the general trend due to the impact of crosstalk. Analyses of C1355 and C1908 (without crosstalk) with independent and fully correlated process variables shows an increase in mean and decrease in standard deviation as we change the correlated structures from fully correlated to independent, as shown in Table 7. This implies that the impact of crosstalk for C1355 and C1908 dominates the impact of changing correlation structures.

**Table 7. Comparison of different correlation structures without crosstalk for C1355 and C1908.**

Circuit	SSTA <sub>full_corr</sub> (ns)		SSTA <sub>zero_corr</sub> (ns)		$\frac{(SSTA_{zero\_corr} - SSTA_{full\_corr})}{SSTA_{full\_corr}}$ (%)	
	Mean	SD	Mean	SD	Mean	SD
C1355	0.75	0.06	0.73	0.07	2.95	-13.08
C1908	0.79	0.04	0.77	0.05	2.47	-9.11

Apart from minor aberrations like the above, similar results were obtained with variations in correlation distances, although the rate of change in the distribution is very slow.

#### *F. Validation of the Linear Crosstalk Model*

The linear crosstalk model is validated by circuit simulations using Cadence Spectre. The circuit consists of two inverters where the first inverter (INV1) has a falling input signal while the second inverter (INV2) has a rising input signal. The two inverters are victim and aggressor to each other. The slew rates were varied over many circuit simulations to verify the crosstalk model. The linear crosstalk model was validated using

other gates as well. The grounded capacitance and the coupling capacitance in the circuit is 1fF. The input slew rate is 20 ps. Figure 18 illustrates the increase in delay due to crosstalk over relative signal arrival time, as simulated with Spectre and predicted by the models.

Our linear crosstalk model overestimates the delay increase by at least 3 times and 4 times for INV1 and INV2 respectively. The reason for this is that the model has a fixed switching window of 20 ps, while the actual switching window is only 10 ps. Similarly, the peak delay is overestimated by the switch factor of two. The peak delay overestimation is due to the different driver strengths in the inverter gate for rising and falling transitions, and also the gates are aggressor to each other. Although, the gate input slew rates are the same, the gate output slew rates are different due to the different transitions.

In many cases, victim and aggressors overlap only during a small portion of the switching window relative to the path delay and due to the normal approximation of the delay increase for each aggressor, the total error in the final path delay distribution is expected to be minimal. Also, in searching for aggressor paths, CodGen generates paths in descending order of delay and hence the longest that is within the switching window will be selected for the corresponding aggressor. Hence, the overestimation in crosstalk delay increase tends to cancel out the underestimation in aggressor path generation.

The increase in delay on INV1 and INV2 are different due to the fact that INV1 has a rising transition at its output while INV2 has a falling transition at its output. Numerical errors are responsible for the variation in the delay increase estimates in the 0-5 fs range. Although our model generates different maximum delay increases for rising

and falling transitions, the difference between the maximum delay increase between rising and falling transitions is negligible and hence a single estimate of the delay increase for both transitions is illustrated in Figure 18. The Relative Signal Arrival Time (RSAT) is calculated at the gate outputs.

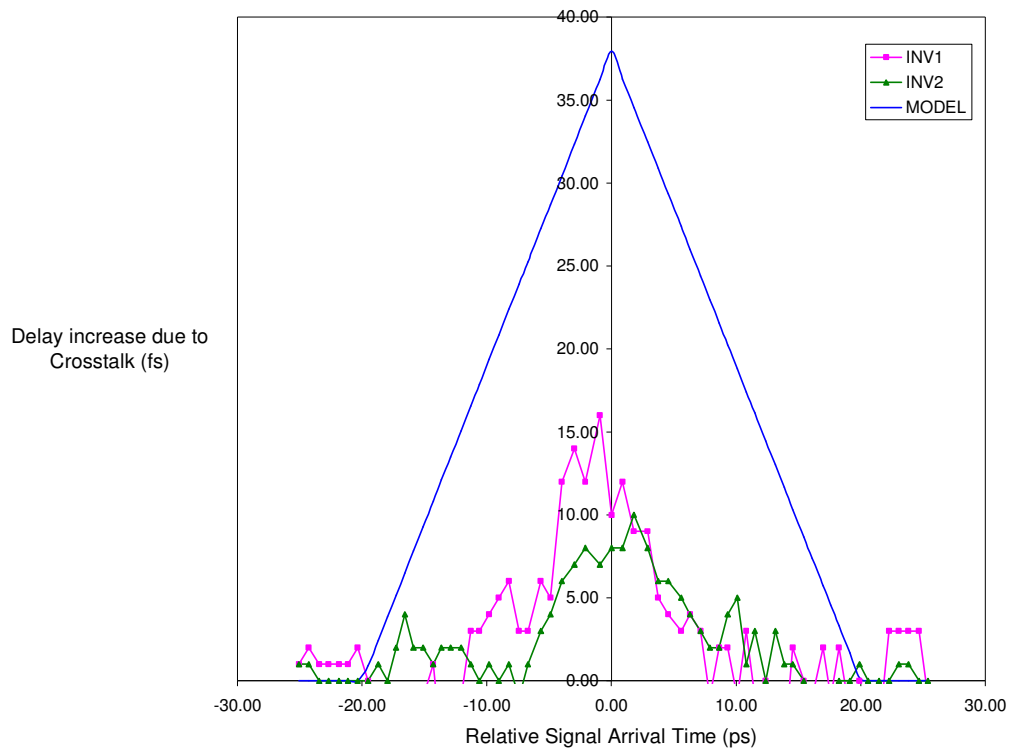


Figure 18. Crosstalk delay increases over relative signal arrival time in a sample circuit with the same input slew rates.

The circuit was simulated with different input slew rates as well. Figure 19 illustrates the delay increase due to crosstalk on the two inverters. INV1 (15 ps) has a higher input slew rate than INV2 (20 ps). A slowly transitioning signal will have a less impact on a faster transitioning signal, while a faster signal will have a larger impact on

the slower signal. Our linear crosstalk model overestimates the delay increase by at least 4 times and 5 times for INV1 and INV2 respectively.

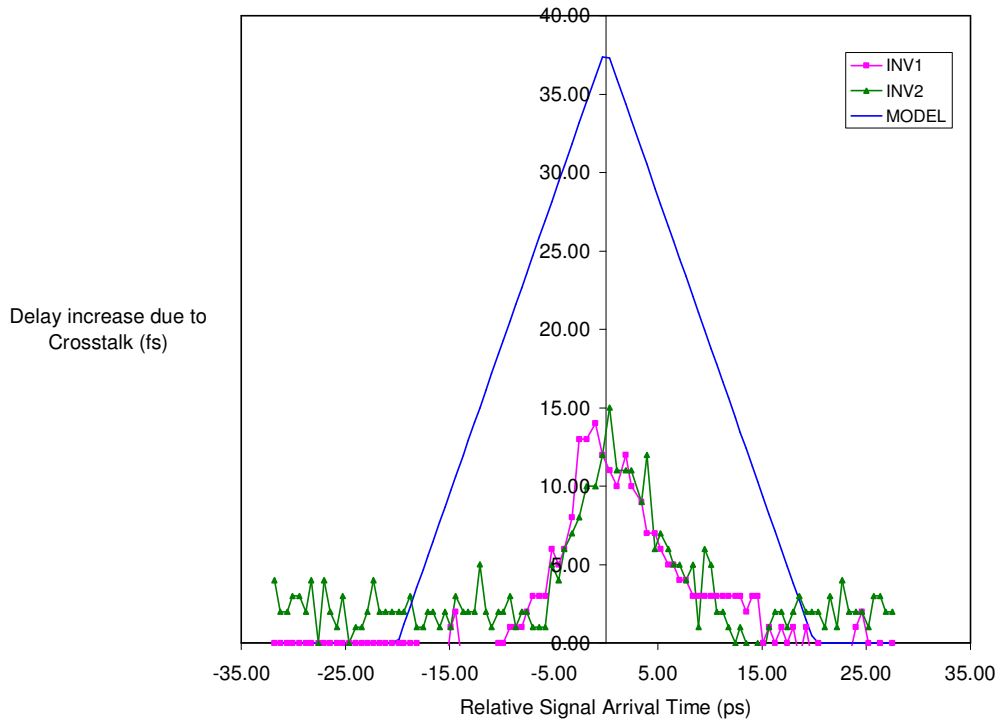


Figure 19. Crosstalk delay increases over relative signal arrival time in a sample circuit with different input slew rates.

Figure 18 and Figure 19 illustrate that a triangle model of approximating delay increase due to crosstalk will fit circuit simulation results. The dimensions of the triangle are determined by the switching factor and switching window size. Although we are overestimating the maximum delay increase due to crosstalk in both cases; the switch factor can be modified by the circuit designer to accurately fit his technology. Similarly, the switching window range can also be selected. The error in overestimating the crosstalk delay increases can be reduced by replacing the crosstalk linear delay increase

function (based on Equation 14) by a new function that takes into account the relative signal arrival time, slew rates, coupling capacitance ratio and the driver strengths.

## V. SUMMARY AND CONCLUSION

A statistical static timing analysis approach has been presented that models crosstalk and spatial correlations in intra-die variations apart from considering gradients and inter-die process variations. The new linear model for crosstalk fits into the domain of statistical static timing analysis very well. It has been shown that an analysis without crosstalk can be quite optimistic. A circuit designer has the option of changing various parameters like correlation function, maximum correlation distance, and grid size which would enable him to use the SSTA tool depending on his time and accuracy tradeoff.

Although the MAX function in its current form worked for seven out of the ten ISCAS 85 circuits, significant errors were noticed for the remaining three circuits. Changing the order of computation of maximum distribution reduced the error for two circuits. It is suspected that a tree approach to compute the maximum of all longest paths could reduce the error in the remaining circuit as well. It is necessary to explore the results of the maximum distribution following a tree approach to do the MAX computation.

The linear model for crosstalk assumes the same slew rates for the victim and the aggressor. But this may not be true always, and hence a better linear crosstalk model that calculates the delay increase due to crosstalk as a function of the delay difference between the victim and the aggressor, slew rates of the victim and the aggressor, and the driver strengths is necessary for more accurate results.

A logical extension of this work would be to increase the speed of path generation, and test the SSTA tool on industrial circuits. The switch factor model which is used to approximate the maximum delay may not be accurate for certain cases and



hence a better model can be incorporated to provide accurate crosstalk delay estimates. Also, an assumption of uniform delay increase for both early aggressor ( $RSAT < 0$ ) and late aggressor ( $RSAT > 0$ ) may be optimistic as in the case of a late aggressor, the crosstalk delay increase falls rapidly and goes to 0 in a short time as evidenced by relative window approaches. The errors in the linear delay model could be minimized by formulating a quadratic delay model that fits into the domain of statistical analysis and crosstalk.

Apart from affecting the delay of the circuit, process variations can also affect the temperature and supply noise of the chip. Preliminary results indicate that the temperature and supply noise may behave like crosstalk in terms of its non-monotonicity. An SSTA tool that models the effect of process variations on delay, temperature and supply noise will be ideal to a circuit designer in his quest to optimally design chips using accurate process variation models.

## REFERENCES

- [1] A. Agarwal, D. Blaauw, V. Zolotov, S. Sundaeswaran, M. Zhao, K. Gala, and R. Panda, "Path-Based Statistical Timing Analysis Considering Inter- and Intra-Die Correlations," in *Proc. Timing Issues in the Specification and Synthesis of Digital Systems*, Sep. 2002, Monterey, California, pp. 16-21.
- [2] A. Devgan and C. Kashyap, "Block-based Static Timing Analysis with Uncertainty," in *Proc. Int. Conf. on Computer-Aided Design*, Nov. 2003, San Jose, California, pp. 607-614.
- [3] H. Chang and S. Sapatnekar, "Statistical Timing Analysis Considering Spatial Correlations Using a Single PERT-like Traversal," in *Proc. Int. Conf. on Computer-Aided Design*, Nov. 2003, San Jose, California, pp. 621-625.
- [4] A. Agarwal, V. Zolotov, and D. T. Blaauw, "Statistical Timing Analysis Using Bounds and Selective Enumeration," *IEEE Trans. on Computer-Aided Design*, vol. 22, no. 9, pp. 1243 - 1260, Sept. 2003.
- [5] C. Visweswariah, K. Ravindran, K. Kalafala, S. G. Walker, and S. Narayan, "First-order Incremental Block-based Statistical Timing Analysis," in *Proc. Design Automation Conf.*, June 2004, San Diego, California, pp. 331-336.
- [6] B. Choi and D. M. H. Walker, "Timing Analysis of Combinational Circuits Including Capacitive Coupling and Statistical Process Variation," in *Proc. VLSI Test Symposium*, Apr. 2000, Montreal, Canada, pp. 49-54.
- [7] J. Le, X. Li, and L. T. Pileggi, "STAC: Statistical Timing Analysis with Correlation," in *Proc. Design Automation Conf.*, June 2004, San Diego, California, pp. 343-348.

- [8] R. Arunachalam, K. Rajagopal, and L. T. Pileggi, "TACO: Timing Analysis with Coupling," in *Proc. Design Automation Conf.*, June 2000, Los Angeles, California, pp. 266-269.
- [9] A. Gattiker, S. Nassif, C. Dinakar, and C. Long, "Timing Yield Estimation from Static Timing Analysis," in *Proc. Int. Symp. on Quality Electron. Design*, Mar. 2001, San Jose, California, pp. 437 - 442.
- [10] *PERT Summary Report, Phase I and Phase II*, Navy Special Project Office, Bureau of Naval Weapons, Navy Department, Washington, D.C., 1958.
- [11] W. Qiu and D. M. H. Walker, "An Efficient Algorithm for Finding the K Longest Testable Paths through Each Gate in a Combinational Circuit," in *Proc. Int. Test Conf.*, Sept. 2003, Charlotte, North Carolina, pp. 592-601.
- [12] M. J. M. Pelgrom, A. C. Duinmaijer, and A. P. G. Welbers, "Matching Properties of MOS Transistors," *IEEE Journal of Solid-State Circuits*, vol. 24, no. 5, pp. 1433-1440, Oct. 1989.
- [13] P. F. Tehrani, S. W. Chyou, and U. Ekambaram, "Sub-Micron Static Timing Analysis in Presence of Crosstalk," in *Proc. Int. Symp. on Quality of Electron. Design*, Mar. 2000, San Jose, California, pp. 505-512.
- [14] F. Dartu and L. T. Pileggi, "Calculating Worst-case Gate Delays due to Dominant Capacitance Coupling," in *Proc. Design Automation Conf.*, June 1997, Anaheim, California, pp. 576-580.
- [15] L. Gal, "On-chip Crosstalk – The New Signal Integrity Challenge," in *Proc. Custom Integrated Circuit Conf.*, May 1995, Santa Clara, California, pp. 251-254.

- [16] "Noise-aware Timing Analysis," Cadence, [Online]. Accessed 27th Apr. 2005; Available from: <http://www.cadence.com/whitepapers/NoiseAware061301.pdf>.
- [17] A. B. Kahng, S. Muddu, and E. Sarto, "On Switch Factor Based Analysis of Coupled RC Interconnects," in *Proc. Design Automation Conf.*, June 2000, Los Angeles, California, pp. 79-84.
- [18] P. Chen, D. A. Kirkpatrick, and K. Keutzer, "Miller Factor for Gate-level Coupling Delay Calculation," in *Proc. Int. Conf. on Computer-Aided Design*, Nov. 2000, San Jose, California, pp. 68-74.
- [19] S. Sapatnekar, "Capturing the Effect of Crosstalk on Delay," in *Proc. Int. Conf. on VLSI Design*, Jan. 2000, Calcutta, India, pp. 364-369.
- [20] S. Hassoun, C. Cromer, and E. Calvillo-Gamez, "Static Timing Analysis for Level-clocked Circuits in the Presence of Crosstalk," *IEEE Trans. on Computer-Aided Design*, vol. 22, no. 9, pp. 1270-1277, Sept. 2003.
- [21] P. D. Gross, R. Arunachalam, K. Rajagopal, and L. T. Pileggi, "Determination of Worst Case Aggressor Alignment for Delay Calculation," in *Proc. Int. Conf. on Computer-Aided Design*, June 1998, San Jose, California, pp. 212-219.
- [22] T. Xiao and M. Marek-Sadowska, "Worst Delay Estimation in Crosstalk Aware Static Timing Analysis," in *Proc. Int. Conf. on Computer Design*, Sept. 2000, Austin, Texas, pp. 115-120.
- [23] P. Chen and K. Keutzer, "Towards True Crosstalk Noise Analysis," in *Proc. Int. Conf. on Computer-Aided Design*, Nov. 1999, San Jose, California, pp. 132-137.

- [24] R. Arunachalam, R. D. Blanton, and L. T. Pileggi, "False Coupling Interactions in Static Timing Analysis," in *Proc. Design Automation Conf.*, June 2001, Las Vegas, Nevada, pp. 726-731.
- [25] T. Yang, J. Kim, J. Choi, M. H. Yoo, and J. T. Kong, "Elimination of False Aggressors Using the Functional Relationship for Full-chip Crosstalk Analysis," in *Proc. Intl. Symp. Quality of Electron. Design*, Mar. 2003, San Jose, California, pp. 344-347.
- [26] Y. Sasaki and G. D. Micheli, "Crosstalk Delay Analysis Using Relative Window Method," in *Proc. ASIC/SOC Conf.*, Sept. 1999, Washington, D.C., pp. 9-13.
- [27] K. Agarwal, Y. Cao, T. Sato, D. Sylvester, and C. Hu, "Efficient Generation of Delay Change Curve for Noise-aware Static Timing Analysis," in *Proc. Asia South Pacific Design Automation Conf./VLSI Design*, Jan. 2002, Bangalore, India, pp. 77-84.
- [28] K. Takeuchi, K. Yanagisawa, T. Sato, K. Sakamoto, and S. Hojo, "Probabilistic Crosstalk Delay Estimation for ASICs," *IEEE Trans. on Computer-Aided Design*, vol. 23, no. 9, pp. 1377-1383, Sept. 2004.
- [29] X. Lu, Z. Li, W. Qiu, W. Shi, and D. M. H. Walker, "Longest Path Selection for Delay Test under Process Variation," in *Proc. Asia South Pacific Design Automation Conference*, Jan. 2004, Yokohama, Japan, pp. 98-103.
- [30] X. Li, P. Gopalakrishnan, X. Yang, and L. T. Pileggi, "Asymptotic Probability Extraction for Non-normal Distributions of Circuit Performance," in *Proc. Int. Conf. on Computer-Aided Design*, Nov. 2004, San Jose, California, pp. 855-862.

- [31] C. E. Clark, "The Greatest of a Finite Set of Random Variables," *Operations Research*, vol. 9, no. 2, pp. 145-162, Mar. 1961.
- [32] S. Tsukiyama, M. Fukui, and M. Tanaka, "A Statistical Static Timing Analysis Considering Correlations between Delays," in *Proc. Asia South Pacific Design Automation Conf.*, Jan. 2001, Yokohama, Japan, pp. 353 - 358.
- [33] E. W. Weisstein, *Normal Distribution*. MathWorld--A Wolfram Web Resource, [Online]. Accessed on 27th Apr. 2005; Available from: <http://mathworld.wolfram.com/NormalDistribution.html>.
- [34] T. Ooura, *Special Functions - Gamma / Error Functions*, [Online]. Accessed on 27th Apr. 2005; Available from: <http://momonga.t.u-tokyo.ac.jp/~ooura/index.html>.
- [35] J. Benkoski, E. V. Meersch, L. J. M. Claesen, and H. D. Man, "Timing Verification Using Statically Sensitizable Paths," *IEEE Trans. on Computer-Aided Design*, vol. 9, no. 10, pp. 1073-1084, Oct. 1990.
- [36] J. A. Bell, "Timing Analysis of Logic-Level Digital Circuits Using Uncertainty Intervals," M.S. thesis, Dept. Comp. Sci., Texas A&M Univ., College Station, 1996.
- [37] P. Goel, "An Implicit Enumeration Algorithm to Generate Tests for Combinational Logic Circuits," *IEEE Trans. on Computers*, vol. C-30, no. 3, pp. 215-222, Mar. 1981.
- [38] F. Brglez and H. Fujiwara, "A Neutral Netlist of 10 Combinational Benchmark Circuits," in *Proc. Int. Symp. Circuits and Systems*, June 1985, Kyoto, Japan, pp. 785-794.

- [39] S. Nassif, "Delay Variability: Sources, Impact and Trends," in *Proc. Int. Solid-State Circuits Conf.*, Feb. 2000, San Francisco, California, pp. 368-369.

## VITA

Senthilkumar Veluswami was born in Erode, India on 16 September 1981. He completed his Bachelor of Engineering degree in computer science and engineering from the College of Engineering, Guindy, Anna University, Chennai, India in May 2003. He joined the graduate school in Texas A&M University as a computer science major in fall 2003 and graduated in summer 2005. His research interests are in VLSI CAD, computer architecture and evolvable hardware. He can be reached by email at [cameosenthil@yahoo.com](mailto:cameosenthil@yahoo.com). His permanent address is c/o. Sridevi Karunanidhi, 2115 Chipstone Road, Charlotte, NC 28262.