STATISTICAL STATIC TIMING ANALYSIS CONSIDERING THE IMPACT OF POWER SUPPLY NOISE IN VLSI CIRCUITS

A Thesis

by

HYUN SUNG KIM

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 2007

Major Subject: Computer Engineering

STATISTICAL STATIC TIMING ANALYSIS CONSIDERING THE IMPACT OF POWER SUPPLY NOISE IN VLSI CIRCUITS

A Thesis

by

HYUN SUNG KIM

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
	Eun Jung Kim
	Weiping Shi
Head of Department,	Valerie Taylor

August 2007

Major Subject: Computer Engineering

ABSTRACT

Statistical Static Timing Analysis Considering the Impact

of Power Supply Noise in VLSI Circuits. (August 2007)

Hyun Sung Kim,

B.S., University of California, Davis

Chair of Advisory Committee: Dr. Duncan Moore Henry Walker

As semiconductor technology is scaled and voltage level is reduced, the impact of the variation in power supply has become very significant in predicting the realistic worst-case delays in integrated circuits. The analysis of power supply noise is inevitable because high correlations exist between supply voltage and delay. Supply noise analysis has often used a vector-based timing analysis approach. Finding a set of test vectors in vector-based approaches, however, is very expensive, particularly during the design phase, and becomes intractable for larger circuits in DSM technology.

In this work, two novel vectorless approaches are described such that increases in circuit delay, because of power supply noise, can be efficiently, quickly estimated. Experimental results on ISCAS89 circuits reveal the accuracy and efficiency of my approaches: in s38417 benchmark circuits, errors on circuit delay distributions are less than 2%, and both of my approaches are 67 times faster than the traditional vector-based approach. Also, the results show the importance of considering care-bits, which sensitize the longest paths during the power supply noise analysis.

DEDICATION

To my parents

ACKNOWLEDGMENTS

I would like to thank my committee chair, Dr. Walker for advising me during my graduate school years at Texas A&M University. His patience, guidance, and assistance have helped me complete the course of this research. I have enjoyed doing my research under his advice.

I would also like to thank my committee members, Dr. Kim, Dr. Mahapatra, and Dr. Shi. They are all great teachers. The materials and ideas they have provided in class have widened my knowledge in architecture as well as VLSI and also helped me address questions in my research work.

I am truly grateful to Kathleen Phillips and Dave Mayes from the department of Agricultural Communications for their financial support during my graduate school years. Besides, they have always been very friendly and encouraging at the work place.

I would sincerely like to thank my parents, brother, and sister for their support and encouragement. In addition, I would like to thank my father-in-law and mother-in-law for their prayers and kindness. Finally, I would like to thank my most beautiful wife, Sara for her love, trust, faith, endurance, care and support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

I. INTRODUCTION

A. Timing Analysis

Timing analysis is a method to predict the performance of integrated circuits. It is frequently used in the design, test, and optimization phases before manufacturing chips. Timing analysis is used to validate the performance of an integrated circuit during the test phase, and to enable performance optimization during the design and optimization phases.

Timing analysis has become very critical in deep submicron (DSM) technology. As semiconductor technology is scaled, the decrease in device feature size, increase in gate density, supply current, and the number of interconnect layers cause process variations, crosstalk noise, leakage current noise and power supply noise. The power supply noise significantly affects circuit timing, but it is not considered in most traditional timing analysis approaches. This problem is challenging because circuit timing affects the power supply noise and vice-versa, and getting worse in newer technologies.

B. Power Supply Noise

In deep submicron (DSM) technology, power supply analysis has become increasingly important in predicting the realistic worst-case delays in integrated circuits [1]. Ideally, the gates in the integrated circuit receive the full supply voltage from the

---

The journal model is *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems.*

power and ground supply networks. However, due to both resistance in the on-chip supply network and inductance in the package interconnect, noise occurs on the power and ground supply networks and affects circuit timing. Fluctuations of 10% in power/ground supply voltage can cause the delay for standard gates to vary by up to 30% in 130 nm technology [2][3]. Tirumurti [4] pointed out that a 1% change in power supply voltage will cause roughly a 4% change in delay for most static CMOS gates at 0.9 V in 90 nm technology. Newer technologies have increased delay sensitivity to supply noise, due to reduced gate overdrive. Therefore, the analysis of power supply noise has become inevitable in timing analysis.

C. Organization of the Thesis

The thesis is organized as follows. Section II provides the background and previous work on timing analysis and power supply analysis. Two proposed *static* technique to estimate supply voltage noise distributions as well as statistical timing analysis with the voltage distributions are explained in Section III. Section IV shows experimental results of the proposed approach on ISCAS89 benchmark circuits. Finally, a conclusion is provided in Section V.

II. BACKGROUND

A. Timing Analysis

Timing analysis is a method to check timing constraints and to calculate slack (spare time) values at primary outputs in the integrated circuit in order to predict the timing performance of the circuit. A slack value can be obtained by subtracting the required arrival time (RAT) from a critical (longest) path delay. A positive slack indicates that the path has timing margin, and can be optimized to increase clock frequency, reduce power dissipation or add functionality. On the other hand, a negative slack value means that the path is slow, and the logic designer must re-design the circuit to reduce the path delay.

Figure 1 illustrates the usefulness of timing analysis during the design optimization phase. By timing analysis, in the given design, timing information before and after optimization is shown. In the bar charts, the values in the left column are slack values and the values in the right column are frequencies of occurrence. In the design before optimization, many negative slack values exist, which implies that the design cannot operate at the desired frequency. In the optimized design, the slack distribution is tightened up, and the worst-case (most negative) slack is significantly reduced.

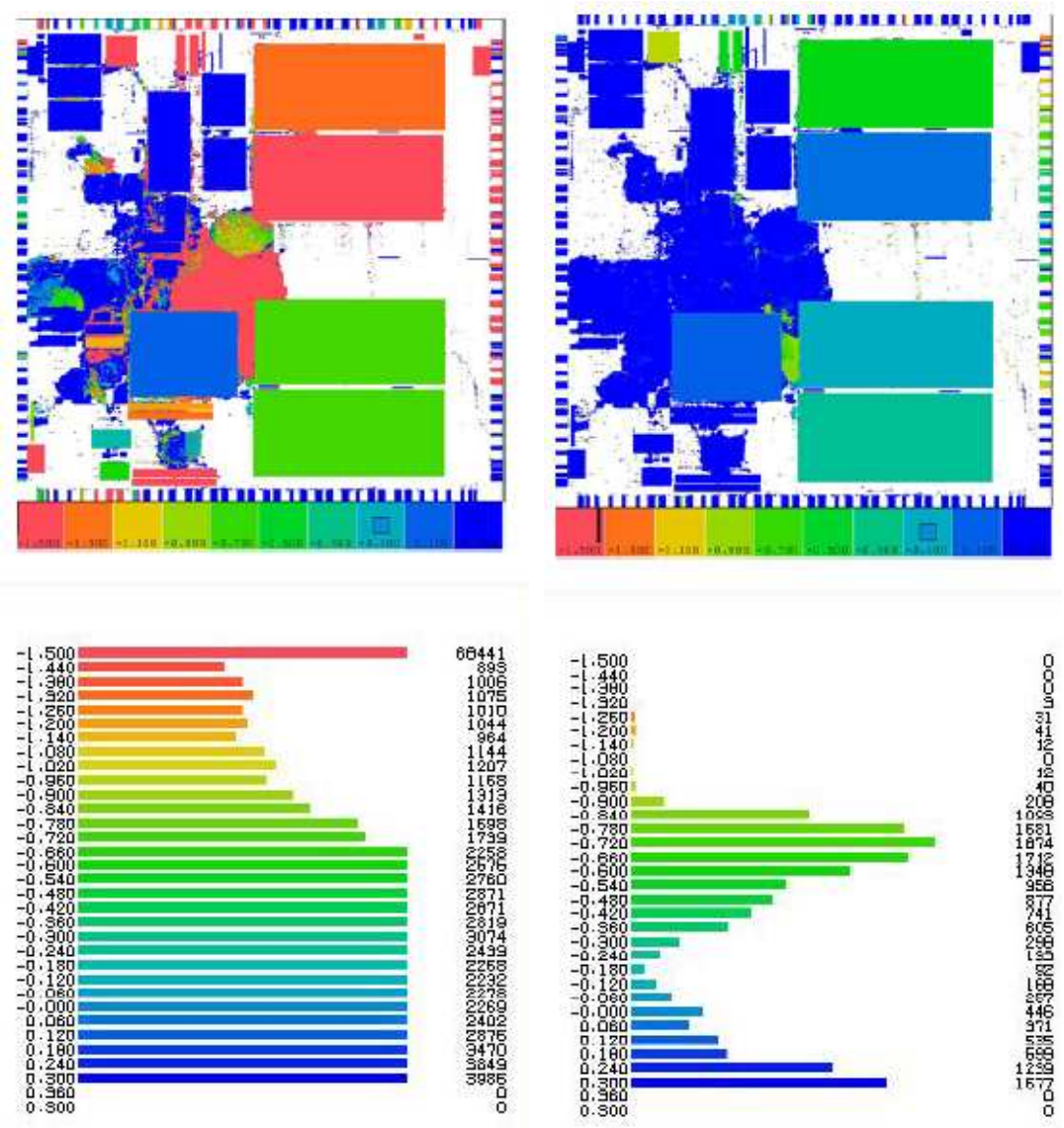

Figure 1. Timing before optimization (left) and after optimization (right) [5]

Timing analysis without a set of (user-defined) input vectors is called *static timing analysis* (STA). STA falls into two broad classes: deterministic static timing analysis and statistical static timing analysis (SSTA). Traditionally, deterministic STA, or

STA, has been used. In STA, process and environmental parameter values are treated as constant, and chosen by a case-based methodology, such as best-case, nominal-case or worst-case. Due to the constant parameter values, *SUM* and *MAX* operations for gate and path delay calculations (Figure 2) are quite easy, as follows:

$$SUM: \quad d_o = d_i + d_k \tag{1}$$

$$MAX: \quad d_o = max\{ d_i + d_k , d_j + d_k \} \tag{2}$$

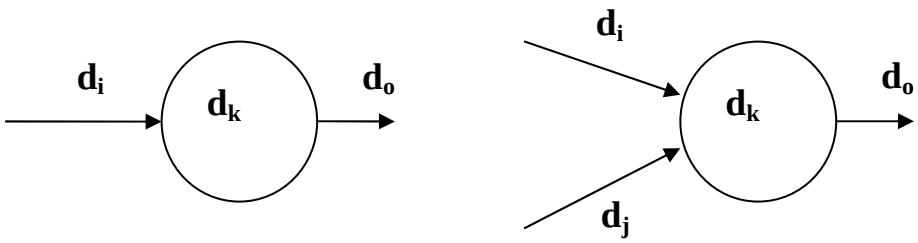

Figure 2. Computation of *sum* and *max* operations

Although the computation for gate and path delays is simple, STA has drawbacks. First, as the number of uncertain parameter values increases, it is intractable to analyze all possible corners of the parameter space [6]. Second, it does not provide information on the likelihood of the design satisfying the given design specifications. Third, the corner-based methodology can be very conservative, since the worst case may

be much worse than the typical case. This can lead to severe over-design in term of area, power, circuit timing, and design time [7]. Finally, corner-based STA computation is very expensive because it requires a large number of corner cases.

Statistical static timing analysis has been proposed to improve the weaknesses of the traditional deterministic static timing analysis. Instead of considering source variations as constant values, SSTA treats them as random variables, which are generally assumed to be Normally distributed (Gaussian) random variables. Signal arrival time and gate delay can be expressed with Cumulative Density Function (CDF) and Probability Density Function (PDF) [8], respectively. Path delay distributions can be calculated by propagating PDFs of gate delays along each of the longest paths with *SUM* operations, and the circuit performance can be computed by applying the *MAX* operation on the PDFs of all path delays, as illustrated in Figure 3. The *SUM* and *MAX* operations will be explained in detail in Section III. SSTA is increasingly favored by designers in DSM technologies because it provides quick, accurate estimations as well as statistical information of circuit performance, which is usefully employed in design, test and optimization phases.

There are two approaches to static timing analysis: *path-based* [9-14] and *block-based* [8][15-20]. The path-based approach can be regarded as a depth-first search, and it employs a given a set of critical paths. While this approach is accurate and can capture correlations, it faces the difficult problem of how to select the set of paths to be tested. The block-based approach uses a breadth-first traversal search technique. This approach is fast, but does not handle correlations due to path sharing. In this work, we will

consider the path-based approach, due to its accuracy.

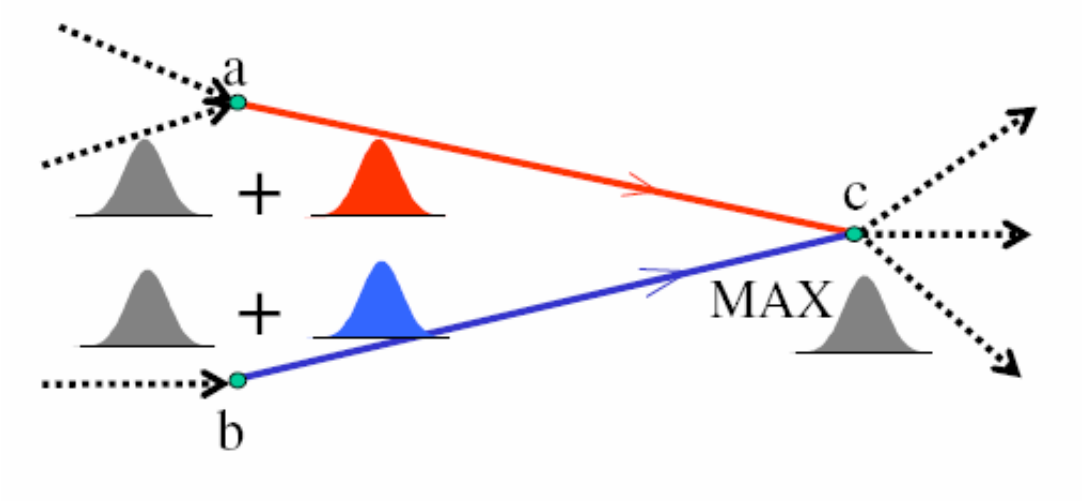

Figure 3. *SUM* and *MAX* operations in SSTA

B. Power Supply Noise

Power supply noise is noise on power supply network, which reduces the ideal power supply voltage. The reduced supply voltage on the gate reduces its drive strength, thereby increasing its delay [2], which may cause increased circuit delay.

Generally, the supply voltage noise is due to both the parasitic resistance (IR) and inductance $(L \cdot di/dt)$ of the on-chip and package interconnect, as illustrated in Figure 4. The on-chip power grid is predominately resistive, with its noise produced by IR drop. Package interconnect has a higher inductance, so its noise is generated primarily by $L \cdot di/dt$ effects. At faster gate switching speeds and higher circuit density,

on-chip inductance must be taken into account on certain nets [3]. In this work, we do not consider on-chip inductance.

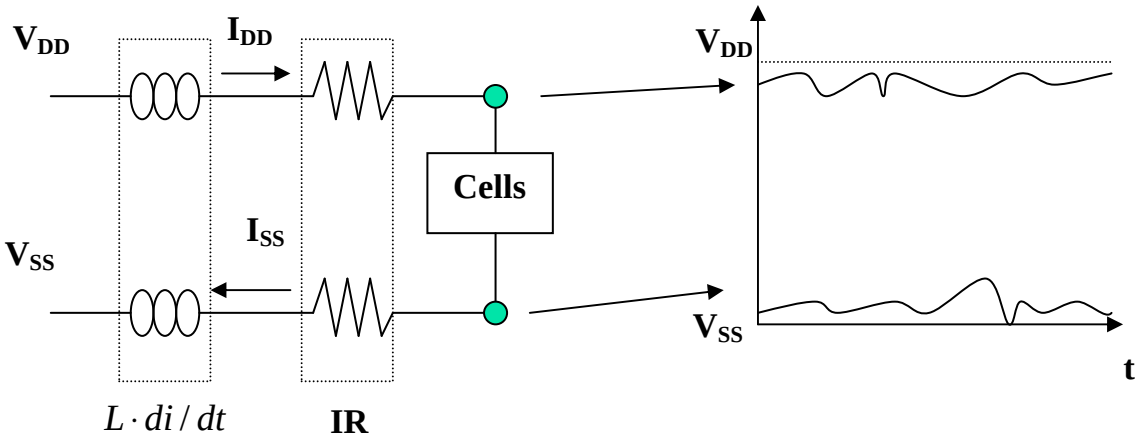

Figure 4. IR and $L \cdot di / dt$ drops on power and ground supply networks

Power supply voltage analysis has been addressed through vector-based and vectorless approaches. Many vector-based approaches [1][21-24] use genetic algorithm (GA) or other optimization algorithms to find a set of input vectors that cause the maximum voltage drop in targeted chip regions. Jiang [1] employed the expensive GA technique to find a set of vectors and applied them to obtain statistical parameters of voltage distributions for all circuit blocks. However, the statistical timing distribution is obtained by iterating their STA, instead of propagating voltage distributions during STA. This method is not practical for large DSM designs. By using a vector-based approach,

Wang [21] proposed a static test vector compaction technique to prevent overkill during delay test. Since Wang was targeting a given set of paths for path delay test, the input vectors had care bits to sensitize the paths. The “don’t care” bits were replaced by two different filling methods: minimum transition filling and random filling. Both [22] and [23] used the genetic algorithm to find test pattern that cause high supply voltage noise. Krstic [24] used two pattern search algorithms: *Timed* ATPG and a probability-based approach. These algorithms produce a small set of patterns that generate a tight low bound on maximum instantaneous currents in the power grid.

Due to the high cost of most vector-based approaches, vectorless supply noise analysis methods were developed [2][25-27]. The vectorless approaches employ circuit timing as well as functional information, a superposition method and supply current constraints. The analysis proposed by Pant [2] is based on a superposition method considering the effects of both voltage drops on individual gates and voltages shifts between driver and receiver gates. The delay maximization on a path due to voltage fluctuation can be computed with supply current constraints, estimated by using Synopsys Powermill, Verilog simulations, or the basis of a previously fabricated part. Bai [25] used sensitivity analysis to express voltages at a supply node in terms of gate currents and then determined the maximum IR voltage drops by solving the linear power grid conductance matrix. In the matrix, gate currents during given intervals of the clock cycle can be computed by using circuit functional relationships. This optimization problem was handled by a constraint graph formulation.

Kouroussis [26] checked the robustness of the power grid by making sure that

the voltage on the power grid does not drop by more than a given threshold. Statistics of voltage drops were computed for this verification by running a sequence of linear programming under a set of user-supplied current constraints, which represent incomplete information on the circuit behavior. Although a Simplex method was employed to speed up the linear programming solver, the analysis took more than 14 hours for a power grid containing less than 2500 nodes. Thus, this approach is not practical in real designs. Kriplani [27] suggested a linear time algorithm (*iMax*) that produced an upper bound envelop of all possible current waveforms for a set of pattern-independent input vectors. Because *iMax* does not produce tight upper bounds and does not handle signal correlations, the *partial input enumeration* (PIE) technique was also proposed to improve the *iMax* technique.

Overall, vector-based approaches are not only very expensive, particularly during early design phase, but also very conservative since the maximum voltage drops generated by a set of input vectors in a local area apply to all cells [2]. In addition, vector-based approaches cannot guarantee worst-case voltage drops using a small set of input vectors [25]. Therefore, the vectorless technique is expected to be attractive for future DSM technology.

We propose two novel vectorless approaches to incorporating supply voltage noise analysis into static timing analysis. These approaches use a set of vectors produced by input pattern generation methods to statistically estimate the realistic power supply noise. We use a supply noise modeling approach developed for delay test generation [21]. Due to the correlation of supply voltage noise between circuit blocks, we adopt the

Principal Component Analysis (PCA) technique [28]. Not only identifies the PCA technique a small set of uncorrelated parameters that explain most of the noise for the circuit blocks, but it also transforms the set of correlated parameters into a set of uncorrelated parameters. Once we obtain all uncorrelated supply voltage variation across the chip, we can determine delay distributions corresponding to the supply voltage distributions for all individual gates on the chip by using a linear delay model [4] and the sensitivity of delay to supply voltage. Then, we can perform the statistical static timing analysis by propagating the individual gate delay distributions along the longest paths on chip. We avoid the abbreviation SSTA, since it is usually associated with statistical process variations, rather than statistical supply voltage variations.

III. SOLUTION METHODOLOGY

A. Power Region & Circuit Switching Model

Because RLC network analysis is expensive, we use a simplified power region model. The maximum voltage drop $\Delta V_{\max}$ in a region during a clock cycle can be estimated with several approximations described in [21]:

$$\Delta V_{\max} = \left(\sum_{i=0}^{n} \int I_{sc\_i} \right) / \left(C_d + C_p \right) \qquad (3)$$

where C_d and C_p are respectively the single lumped decoupling capacitance and the total parasitic capacitances of devices and interconnect connected to the power supply network in a region. The switching current, denoted as $\sum_{i=0}^{n} \int I_{sc\_i}$, is the summation of currents that flow into all *n* switching gates in the region during the clock cycle.

We also employ the circuit switching model, which is similar to the approximated models proposed in [3][4], to estimate the switching current. The switching current consists of leakage current and charging/discharging current. We do not discuss leakage current further, since we treat it as constant, and analyze its voltage impact with a one-time IR drop analysis. As shown in Figure 5, switching current is approximated by a piecewise linear current waveform, which is triangular for small load capacitances and trapezoidal for large load capacitances.

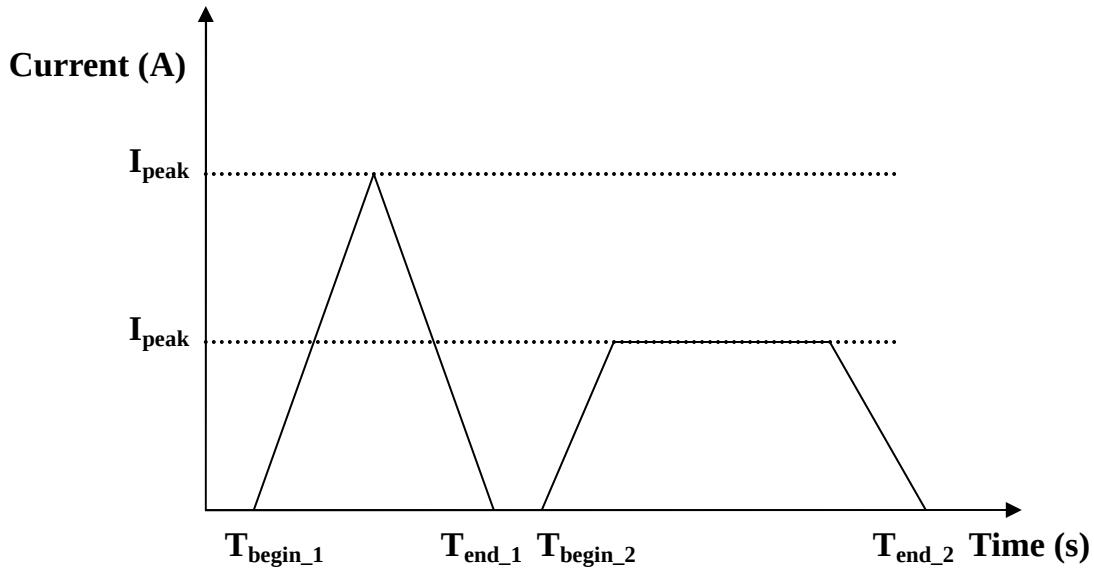

Figure 5. Triangular and trapezoidal current waveforms

B. Statistical Model for Supply Voltage Noise

Given a netlist and circuit initialization, we can group adjacent cells into blocks in the circuit. We run zero-delay logic simulation on the circuit with three different types of input pattern generation methods: MC (Monte Carlo approach), AAC (Analytical Approach with Care bits), and AAR (Analytical Approach with Random bits). We first generate the top 200 longest paths in the circuit and the corresponding sensitizing patterns in each benchmark circuit by using the CodGen ATPG tool [29]. In the Monte Carlo approach, all “don’t care” bits in the sensitizing patterns are randomly filled, and in AAR all care bits as well as “don’t care” bits are filled with randomly generated bits.

In AAC, one set of care bits is selected as follows: we choose the path, among the 200 longest paths, which had the highest probability of being the critical path among 1000 Monte Carlo samples, and use the care bits of its sensitizing pattern for all the input patterns. The “don’t care” bits are then randomly filled.

The set of input patterns in AAR and AAC is simulated to obtain statistical parameters of supply voltage noise distributions for each block. We assume that the random variables for each block are Gaussian random variables. Because of correlations in voltage noise distributions between blocks, we employ the PCA technique. The PCA method transforms a set of correlated random variables $\vec{X} = \{x_1, x_2, ..., x_n\}$ with a covariance matrix *M* into a set of uncorrelated random variables $\vec{X}' = \{x_1', x_2', ..., x_n'\}$, such that any random variable $x_i \in \vec{X}$ can be expressed as a linear function of the principal components with 0 mean and 1 variance in $\vec{X}'$:

$$x_i = \mu_i + \left(\sum_j \sqrt{\lambda_j} \cdot v_{ij} \cdot x'_j \right) \cdot \sigma_i \qquad (4)$$

where μ_i and σ_i are the mean and the standard deviation of x_i, λ_j is the j^{th} eigenvalue of the covariance matrix *M*, v_{ij} is the i^{th} element of the j^{th} eigenvector of *M* and $x'_j \in \vec{X}'$ [16]. The PCA technique is incorporated in both statistical voltage noise analysis and statistical timing analysis for quick, efficient computation.

C. Statistical Static Timing Analysis with Power Supply Noise Variation

With the statistical parameters from the fast power supply noise analysis, we can

also statistically evaluate the performance of the circuit. Here, we consider temporal and spatial supply voltage noise variation.

C.1. Temporal and Spatial Voltage Variation

Because power supply noise and logic gate switching times are both uncertain, it is very difficult to determine the supply voltage at the time the output of a logic gate switches. We adopt the approximation proposed by Wang [21], using the average of the initial and worst-case supply voltages during the clock cycle, where the worst-case supply voltages can be computed with input vectors at each transition on gates. Thus, a maximum voltage drop at each cycle can be calculated as follow:

$$V_{cycle} = V_{initial} - \frac{1}{2} \cdot \Delta V_{max} \qquad (5)$$

In addition to temporal variation, supply voltage has spatial variation. If driver and receiver gates are far enough apart, they can have different supply voltages on them. This can significantly affect the gate delay because the charging/discharging current heavily depends on the input supply voltage. Hashimoto [30] proposed PG (power-ground) level equalization – after equalizing input supply voltage and gate supply voltage, the output load capacitance is increased/decreased by the same ratio. However, we found that Hashimoto's method does not work well over the range of output loads and input slopes. We obtained results that are more accurate by equalizing the input and gate supply voltage without changing the output load capacitance.

C.2. Individual Gate Delay Model

We employ the gate delay model proposed in [30] to calculate the gate delay and output transition time:

$$t_d = f\left(t_{in},\ C_{load},\ V_\mu\right) \tag{6}$$

$$t_{out} = g\left(t_{in},\ C_{load},\ V_\mu\right) \tag{7}$$

where t_{in} and t_{out} are the input and output transition time, respectively, t_d is the gate delay, C_{load} is the output load capacitance and V_μ is the mean of the cycle-to-cycle receiver supply voltage V_{rv_cycle}. Since the supply voltage is a random variable, we utilize the sensitivity of supply voltage versus delay to compute an individual gate delay distribution.

$$\delta = \frac{\left(f\left(t_{in}, C_{load}, V_\mu\right) - f\left(t_{in}, C_{load}, V_{\mu+\sigma}\right)\right)}{\left(V_{\mu+\sigma} - V_\mu\right)} \tag{8}$$

where V_μ and $V_{\mu+\sigma}$ are the mean and (mean + standard deviation), respectively, of the time-varying supply voltage V, and δ is the sensitivity of delay versus supply voltage.

C.3. Computation of Path Delay Distributions

Once individual gate delay random variables are computed, we use the *SUM* operation from the PCA properties in order to calculate path delay distributions as

follows: Let d_i and d_j be two delay random variables on gate *i* and *j* on a path, respectively.

$$d_i = \mu_i + \alpha_1 \cdot x_1 + \alpha_2 \cdot x_2 + \cdots + \alpha_{n-1} \cdot x_{m-1} + \alpha_n \cdot x_m \qquad (9)$$

$$d_j = \mu_j + \beta_1 \cdot x_1 + \beta_2 \cdot x_2 + \cdots + \beta_{n-1} \cdot x_{m-1} + \beta_n \cdot x_m \qquad (10)$$

where μ_i and μ_j are mean delay values, $\sum_{k=1}^{n} \alpha_k$ and $\sum_{k=1}^{n} \beta_k$ are coefficient values of principal components, and $\sum_{k=1}^{m} x_k$ are principal component variables with zero mean and unit variation. Then, the sum of the two delay random variables can be computed as follows:

$$SUM(d_i, d_j) = \mu_i + \mu_j + \sum_{k=1,l=1}^{k=n,l=m} (\alpha_k + \beta_k) \cdot x_l \qquad (11)$$

So, the mean delay $\mu_{SUM} = \mu_i + \mu_j$, and the variation $\sigma_{SUM}^2 = \sum_{k=1}^{k=n} (\alpha_k + \beta_k)^2$.

C.4. Computation of Circuit Delay Distributions

After calculating delay distributions for each given the longest path using the *SUM* operation, we can estimate the circuit delay distribution by the *MAX* operation; that is,

Circuit Delay Distribution = *MAX* (All path delay distributions). (12)

Computing the maximum of all path delay distributions can be done in two different pair-wise structures: unsorted structure and tree-like structure. In the unsorted structure, first we compute the maximum of the delay distributions of the first two paths. This process is repeated, using the new distribution and that of the next path. This process is illustrated in Figure 6.

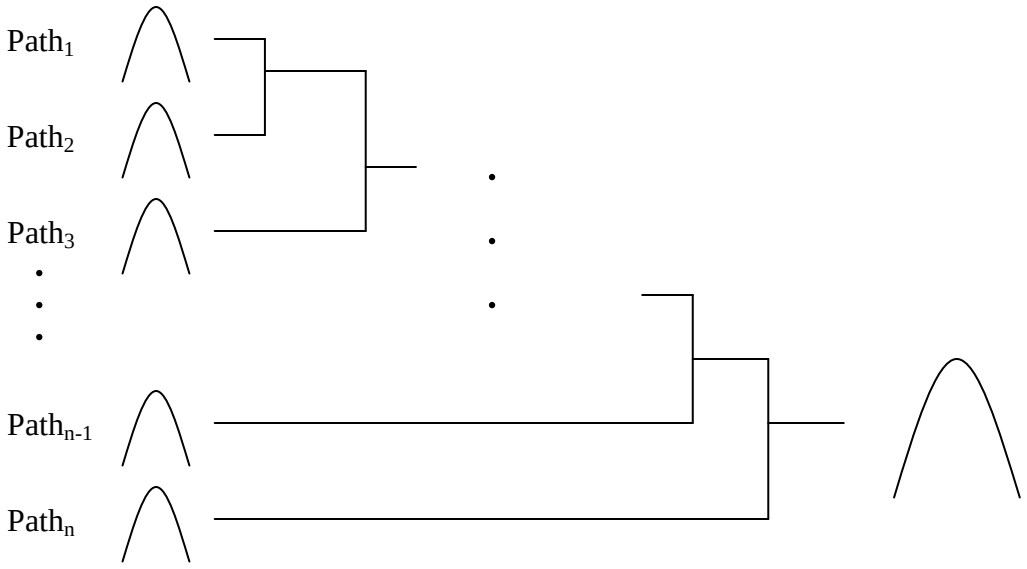

Figure 6. Unsorted Structure in *MAX* operation

The second alternative pair-wise structure we employ is a tree structure. In the tree structure, we first sort all path delay distributions based on their nominal delay values. Then, we calculate maximum delay distributions from the first two paths, the third and fourth paths, and so on applied. We iterate these steps until we find one final

maximum delay distribution. Figure 7 illustrates the procedure.

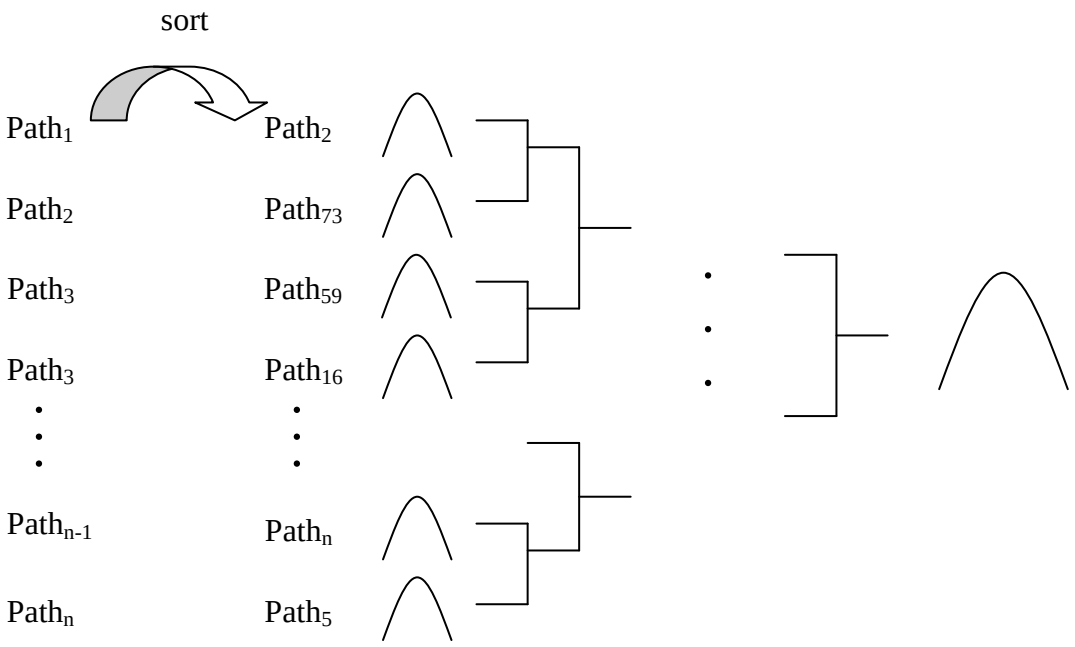

Figure 7. Tree Structure in *MAX* operation

In the unsorted and the tree structures, the *MAX* operation is used to compute the circuit delay distribution. The *MAX* operation is a pair-wise function explained in detail as follows [31]:

1) Compute the means and standard deviations of d_i and d_j.
2) If one of standard deviations is zero, stop here and the distribution with the larger mean becomes the approximation of d_{max}.
3) Compute the correlation between d_i and d_j by using

$$\text{correlation}_{ij} = \rho_{ij} = \frac{\text{covariance}_{ij}}{\sigma_i \cdot \sigma_j} \tag{13}$$

$$\text{where} \quad \text{covariance}_{ij} = \sum_{k=1}^{n} \alpha_k \cdot \beta_k \tag{14}$$

from the PCA properties.

4) If the correlation is one and $\sigma_i = \sigma_j$, stop here and the distribution with a larger mean value becomes the approximation of d_{max}.

5) Compute $\mu_{d_{\max}}$ and $\sigma^2_{d_{\max}}$ as follows:

$$\mu_{d_{\max}} = \mu_{d_i} \cdot \Phi(\beta) + \mu_{d_j} \cdot \Phi(\beta) + \alpha \cdot \varphi(\beta) \tag{15}$$

$$\sigma^2_{d_{\max}} = \left(\mu^2_{d_i} + \sigma^2_{d_i}\right) \cdot \Phi(\beta) + \left(\mu^2_{d_j} + \sigma^2_{d_j}\right) \cdot \Phi(-\beta) + \left(\mu_{d_i} + \mu_{d_j}\right) \cdot \alpha \cdot \varphi(\beta) - \mu^2_{d_{\max}} \tag{16}$$

$$\text{, where} \quad \alpha = \sqrt{\sigma^2_{d_i} + \sigma^2_{d_j} - 2 \cdot \sigma_{d_i} \cdot \sigma_{d_j} \cdot \rho_{ij}} \tag{17}$$

$$\beta = \frac{\mu_{d_i} - \mu_{d_j}}{\alpha} \tag{18}$$

$$\varphi(x) = \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{x^2}{2}\right] \tag{19}$$

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \exp\left[-\frac{y^2}{2}\right] \cdot dy \tag{20}$$

6) Compute coefficient values $a_1, a_2, ..., a_{n-1}, a_n$ of corresponding principal component $x_1, x_2, ..., x_{n-1}, x_n$ in d_{max} by using correlation coefficient equation

$$\gamma(d_{\max}, x_k) = \frac{\sigma_{d_i} \cdot \rho_{d_i\, d_i - x_k} \cdot \Phi(\beta) + \sigma_{d_j} \cdot \rho_{d_j\, d_j - x_k} \cdot \Phi(-\beta)}{\sigma_{d_{\max}}} \tag{21}$$

and $\gamma(d_{\max}, x_k) = \dfrac{a_k}{\sigma_{d_{\max}} \cdot 1}$ from PCA properties.

Thus, $a_k = \sigma_{d_i} \cdot \rho_{d_i\, d_i - x_k} \cdot \Phi(\beta) + \sigma_{d_j} \cdot \rho_{d_j\, d_j - x_k} \cdot \Phi(-\beta)$ (22)

7) Normalize coefficient values as follows:

$$a_k = a_k \frac{\sigma_{d_{\max}}}{S_o} \tag{23}$$

$$\text{where} \quad S_o = \sqrt{\sum_{l=1}^{n} a_l} \tag{24}$$

By repeating the above *MAX* operation of two random variables, we can compute an overall circuit delay distribution from the multiple random variables.

IV. RESULTS

A. Implementation Details

The experiments use ISCAS89 benchmark circuits implemented in 1.8V, 180nm static CMOS technology. We use the CodGen ATPG tool to generate a set of longest paths and corresponding set of path-dependent input patterns. The input patterns consist of “don’t care” bits and care-bits, where the care-bits sensitize the longest paths.

B. Validation of Power Region and Circuit Switching Models

In our first experiment, we first validate the accuracy of the Monte Carlo approach, denoted as MC, which employs the simplified power grid and gate delay models, by comparing results with Cadence *Spectre* simulation, denoted as SS. For validation, we select the path with the highest probability of being the longest from benchmark s1488. All “don’t care” bits in the sensitizing pattern are randomly filled. A set of these randomly filled patterns is generated, and simulated, to obtain the simulated cycle-to-cycle voltage drop distribution.

Figure 8 shows the distributions of supply voltage drops across circuit s1488 using the MC and SS methods. Whereas the SS voltage drop distribution is approximately Normal, the MC distribution is not. One possible reason the MC distribution is not Normal is the very small number of gates (673) in s1488. Figure 9 shows that the MC distribution of cycle-to-cycle voltage drops in s38417, a much larger circuit, is close to Normal. Although the means of the MC and SS voltage drop

distributions in s1488 are close, there is large difference in their standard deviations. The simplified power region, approximated circuit switching models, and the small number of gates in s1488 are possible explanations for this difference.

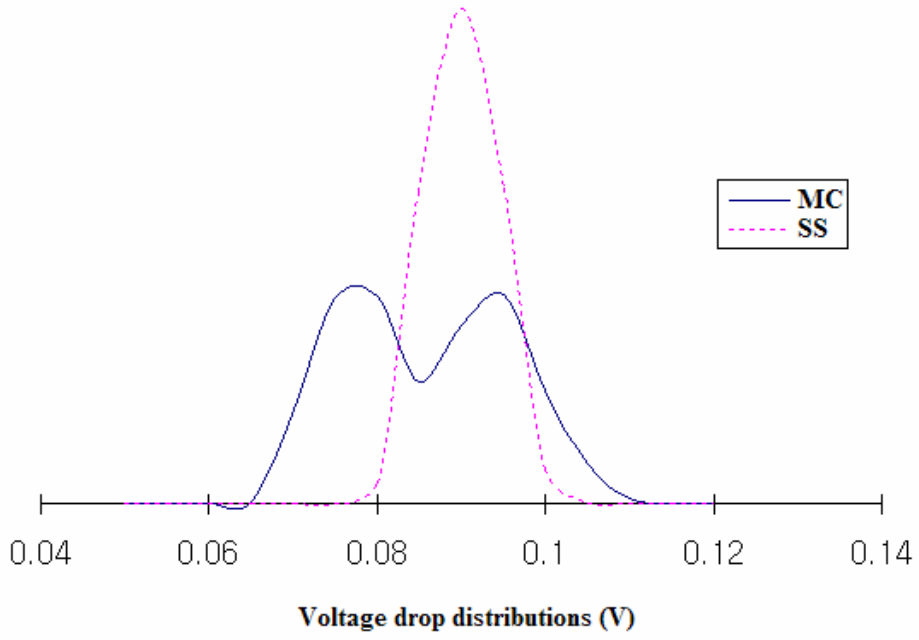

Figure 8. Voltage drop distributions of MC and SS in s1488

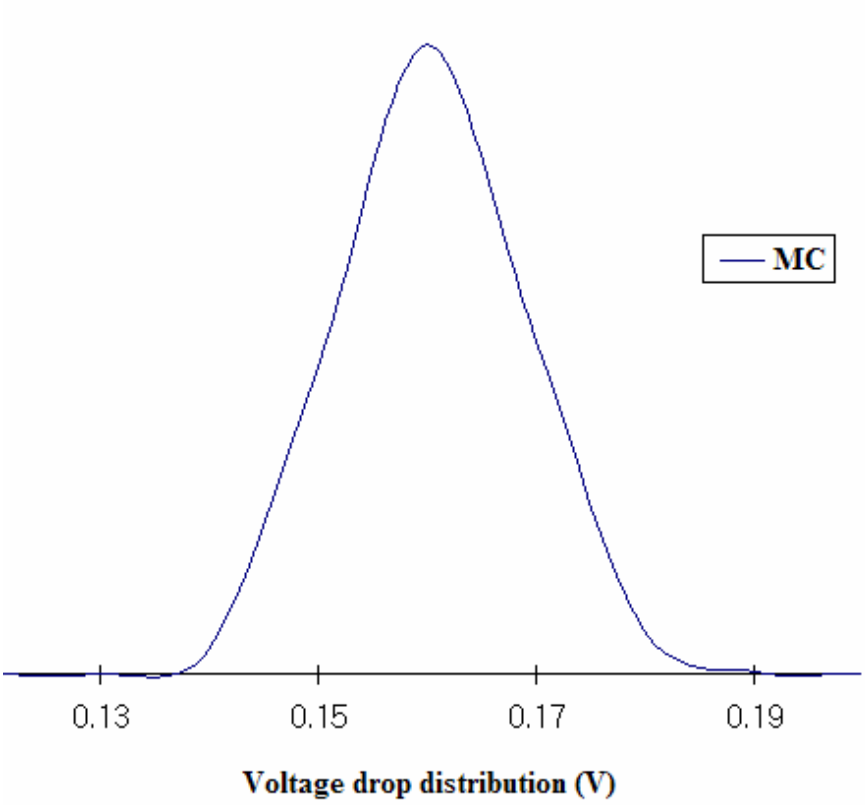

Figure 9. A voltage drop distribution in s38417

As shown in Figure 10, we compute the realistic worst-case cycle-to-cycle delay distribution of the longest path in s1488 circuit, using the voltage noise computed using the MC and SS approaches. The differences between the MC and SS worst-case delay distributions are much smaller than those between the voltage drop distributions in Figure 8. In other words, the differences in the standard deviations of voltage drop distributions in Figure 8 have little impact on the delay distributions. This may be because of the relatively low sensitivity between delay and supply voltage in the 180 nm

technology, or due to an averaging effect. We will further investigate this in much larger benchmark circuits in the future. The results in Figure 8 and Figure 10 are summarized in Table 1.

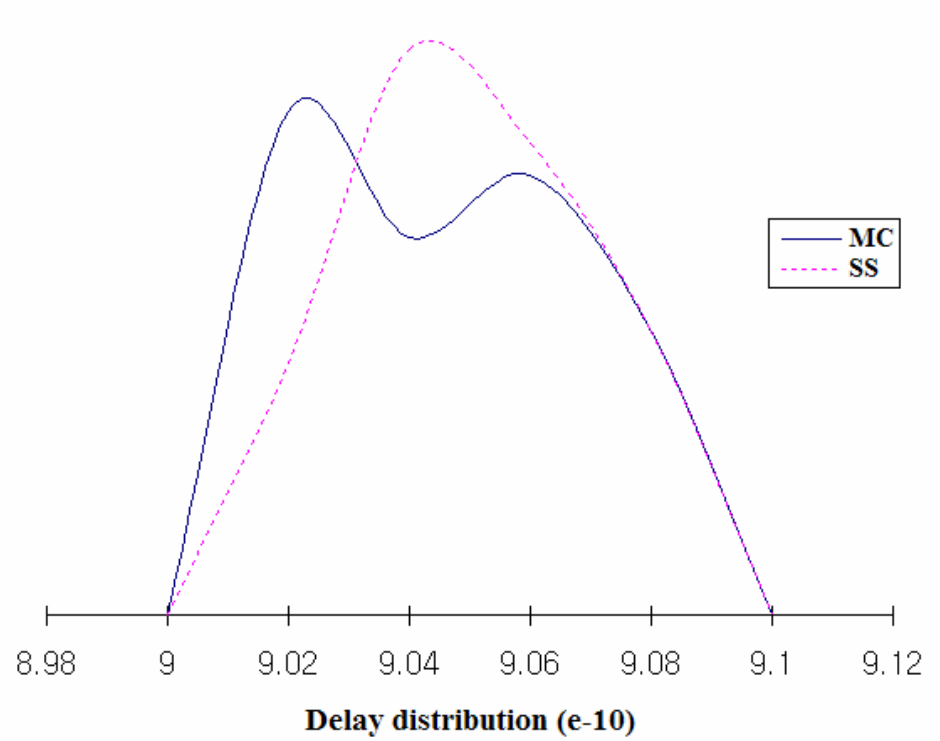

Figure 10. Path delay distributions in MC and SS

Table 1. Voltage drops and delays in s1488

Circuit s1488	MC		SS		Error	
	Voltage drop (V)	Delay (ps)	Voltage drop (V)	Delay (ps)	Due to Voltage (%)	Due to Delay (%)
μ	0.083	904	0.088	904	5.68	0
σ	0.010	2.21	0.004	1.82	60	17.6

C. Monte Carlo vs. Two Proposed Approaches

In our second experiment, we apply the MC calculation for a large number of input patterns, to compute the circuit delay distribution. Circuit simulation was too expensive to generate this large number of samples. We then compare these results to two proposed analytical approaches. Unlike the first experiment, we use the 200 longest paths in each circuit, and group physically adjacent cells into blocks. Here, we use nine blocks in a 3x3 pattern for each benchmark circuit.

The MC approach was used to perform a voltage noise analysis and static timing analysis with the top 200 longest paths in each benchmark circuit. In the voltage analysis, “don’t care” bits in each input pattern were randomly filled. With 1000 different random fill patterns for each path, we compute the delay distributions for each path. After that, we combine the 200 delay distributions using a pair-wise numerical *MAX* function to obtain the MC maximum delay distribution for the circuit. The numerical *MAX* function avoids the error of the analytical *MAX* function described in Section C.4, but is too

expensive for normal use. Because the Monte Carlo approach is too expensive for normal use as a static timing analysis approach, we developed two types of fast analytical approaches: with and without considering the care-bits required to sensitize the longest circuit paths, denoted as AAC and AAR, respectively.

AAC first performs a statistical voltage noise analysis (as described in Section II), with care bits. We then use PCA to transform the set of correlated voltage random variables across regions on the chip into a set of uncorrelated voltage random variables. Given the uncorrelated voltage random variables, we employ the gate delay model as well as the sensitivity model to compute the gate delay distribution. Then, we propagate all gate delay distributions using the *SUM* operation along the 200 longest paths. Finally, we get circuit performance by applying the analytical *MAX* operation (Section C.4) to all 200 delay distributions. Since we do voltage noise analysis and timing analysis statistically, we only need to perform this analysis once for each benchmark circuit. That is why this approach is very fast when compared to MC. The AAR approach is identical to AAC, except that all input bits are random.

Table 2 shows the means and standard deviations of cycle-to-cycle delay distributions calculated by MC, AAC, and AAR. The μ and σ from AAR for circuit s1488 are farther off than for much larger circuits s35932 and s38417. As with the first experiment, this may be due to the small size of s1488. It can be seen that using random inputs rather than longest-path care bit values causes an underestimate in mean delay and overestimate in standard deviation.

We see a similar, but less severe phenomenon in s35932. There is little

difference in mean delay, but AAC reduces the error in σ by 36 % compared to AAR. This is surprising considering that only 0.2% of the input bits are care bits. This small number of bits causes a noticeable change in supply noise, and delay variation.

Unlike in s1488 and s35932, AAR results for the worst case delay μ and σ match that of MC in circuit s38417. Figure 11 illustrates the accuracy of the analytical approaches versus the numerical approach in s38417. One reason for the accuracy in s38417 may be that the larger number of gates causes the noise to appear more random, and the longer paths causes more averaging. The other reason may be that the care bits for the longest paths in this design have less impact on supply noise. Note that even though the noise-induced delay variation in Figure 11 is small, this is only due to the robust power grid design, and does not affect the accuracy of the analysis technique.

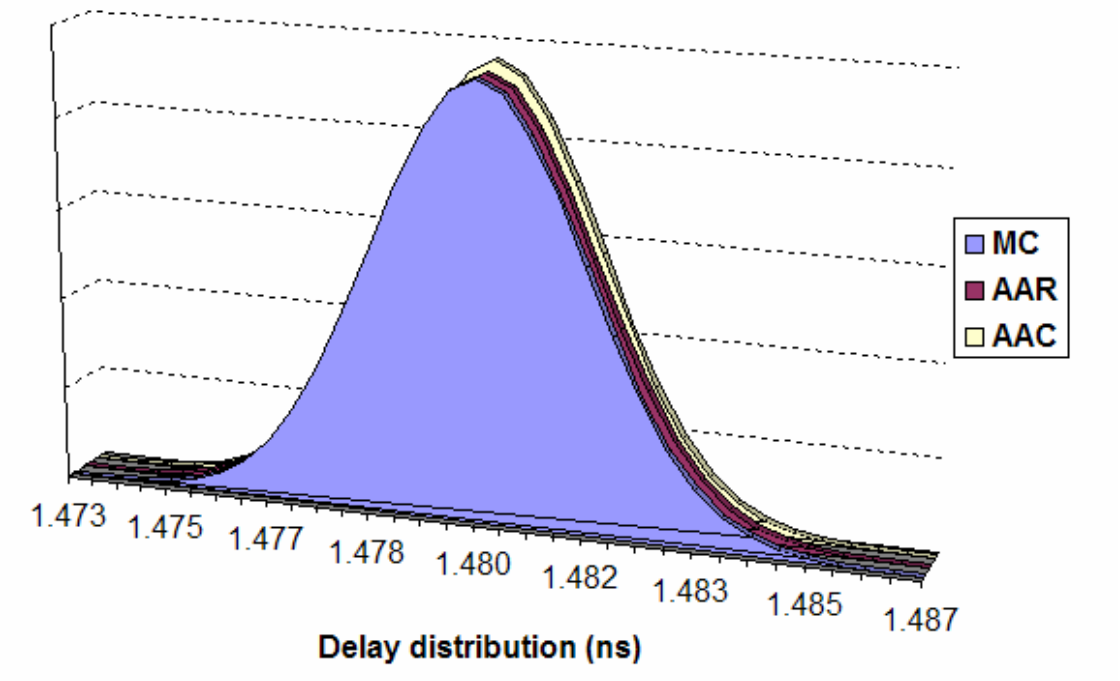

Figure 11. Delay distributions in s38417

In s35932 and s38417, we notice that the σ values for AAC are still off from the σ values for MC. This is because there are multiple paths with significant probability of being the longest path for any one random pattern. The standard deviation errors can be reduced by intelligently deciding which care bits should be used.

Table 2. Statistical parameters of delay distributions in MC, AAR, AAC

Benchmarks	MC			AAR			AAC		
	μ (ps)	σ (ps)	CPU Time	μ (ps)	σ (ps)	CPU Time	μ (ps)	σ (ps)	CPU Time
s1488	927	3.20	264 s	901	11.2	11 s	927	3.06	11 s
s35932	336	0.36	1.54 hr	335	1.03	107 s	336	0.26	107 s
s38417	1480	1.69	2.52 hr	1480	1.69	134 s	1480	1.66	134 s

Finally, we observe that both AAR and AAC are much faster than MC. Statistical power noise analysis and timing analysis reduce run-time. Thus, the analytical approaches can be very helpful for quickly estimating the impact of supply noise during early design phases.

V. SUMMARY AND CONCLUSION

In DSM technology, power supply noise analysis must be performed during timing analysis. Supply noise analysis has often used a vector-based approach. However, this is very expensive, particularly during the early design phase. In this research, we introduced novel vectorless approaches, with and without considering care-bits, which sensitize the longest paths in the circuit. These methodologies can be used efficiently and accurately to estimate the delay increases due to power supply noise. Our experiments on ISCAS89 circuits also demonstrate the importance of a small number of care-bits during the power supply noise analysis.

Future directions for this research are to extend the analysis to large hierarchical chips. The spatial and temporal correlation structure of the supply noise is more complex in such cases, particularly with clock and power gating. In such situations, the supply noise between modules will look less random than between the gates within a module.

REFERENCES

[1] Y. M. Jiang and K. T. Cheng, “Analysis of Performance Impact Caused by Power Supply Noise in Deep Submicron Devices,” *ACM/IEEE Design Automation Conf.*, New Orleans, LA, June 1999, pp. 760-765.

[2] S. Pant, D. Blaauw, V. Zolotov, S. Sundareswaran and R. Panda, “Vectorless Analysis of Supply Noise Induced Delay Variation,” *IEEE/ACM Int’l Conf. Computer Aided Design*, San Jose, CA, Nov. 2003, pp. 184-191.

[3] H. H. Chen and D. D. Ling, “Power Supply Noise Analysis Methodology for Deep Submicron VLSI Chip Design,” *ACM/IEEE Design Automation Conf.*, Anaheim, CA, June 1997, pp. 638-643.

[4] C. Tirumurti, S. Kundu, S. Sur-Kolay and Y.-S. Chang, “A Modeling Approach of Addressing Power Supply Switching Noise Related Failures of Integrated Circuits,” *Design Automation and Test in Europe Conference and Exhibition*, Paris, France, Feb. 2004, pp. 1078-1083.

[5] J. Vygen, “Placement – The Key Problem in Physical Design,” Tutorial at *Int’l Conf. Computer Aided Design*, San Jose, CA, Nov. 2002, pp. 1-121.

[6] C. Visweswariah, K. Ravindran, K. Kalafala and S. Narayan, “First-Order Incremental Block-Based Statistical Timing Analysis,” *ACM/IEEE Design Automation Conf.*, San Diego, CA, June 2004, pp. 331-336.

[7] D. Sinha, N. V. Shenoy and H. Zhou, “Statistical Gate Sizing for Timing Yield Optimization,” *IEEE/ACM Int’l Conf. Computer Aided Design*, San Jose, CA,

Nov. 2005, pp. 1037-1041.

[8] A. Devgan and C. Kashyap, "Block-based Static Timing Analysis with Uncertainty," *IEEE/ACM Int'l Conf. Computer Aided Design*, San Jose, CA, Nov. 2003, pp. 607-614.

[9] A. Agarwal, D. Blaauw, V. Zolotov, S. Sundarswaran, M. Zhao, K. Gala, and R. Panda, "Path-Based Statistical Timing Analysis Considering Inter- and Intra-Die Correlations," *IEEE/ACM International Workshop on Timing Issues in the Specification and Synthesis of Digital Systems*, Monterey, CA, Sep. 2002, pp. 16-21.

[10] A. Gattiker, S. Nassif, C. Dinakar, and C. Long, "Timing Yield Estimation from Static Timing Analysis," *Int'l Symp. on Quality of Electronic Design*, San Jose, CA, Mar. 2001, pp. 437-442.

[11] C. S. Amin, N. Menezes, K. Killpack, F. Dartu, U. Choudhury, N. Hakim and Y. I. Ismail, "Statistical Static Timing Analysis: How Simple Can We Get?," *ACM/IEEE Design Automation Conf.*, Anaheim, CA, June 2005, pp. 652-657.

[12] J. Liou, A. Krstic, L. Wang and K. Cheng, "False-path-aware Statistical Timing Analysis and Efficient Path Selection for Delay Testing and Timing Validation," *ACM/IEEE Design Automation Conf.*, New Orleans, LA, June 2002, pp. 566-569.

[13] J. A. G. Jess, K. Kalafala, S. R. Naidu, R. H. J. Otten and C. Visweswariah, "Statistical Timing for Parametric Yield Prediction of Digital Integrated Circuits," *ACM/IEEE Design Automation Conf.*, Anaheim, CA, June 2003, pp. 932-937.

[14] M. Orshansky and A. Bandyopadhyay, “Fast Statistical Timing Analysis Handling Arbitrary Delay Correlations,” *ACM/IEEE Design Automation Conf.*, San Diego, CA, June 2004, pp. 337-342.

[15] J. Le, X. Li, and L. T. Pileggi, “STAC: Statistical Timing Analysis with Correlation,” *ACM/IEEE Design Automation Conf.*, San Diego, CA, June 2004, pp. 343-348.

[16] H. Chang and S.S. Sapatnekar, “Statistical Timing Analysis Considering Spatial Correlations Using A Single Pert-like Traversal,” *IEEE/ACM Int’l Conf. Computer Aided Design*, San Jose, CA, Nov. 2003, pp. 621-625.

[17] J. J. Liou, K. T. Cheng, S. Kundu and A. Krstic, “Fast Statistical Timing Analysis by Probabilistic Event Propagation,” *ACM/IEEE Design Automation Conf.*, Las Vegas, NV, June 2001, pp. 661-666.

[18] A. Agarwal, D. Blaauw, V. Zolotov and S. Vrudhula, “Computation and Refinement of Statistical Bounds on Circuit Delay,” *ACM/IEEE Design Automation Conf.*, Anaheim, CA, June 2003, pp. 348-353.

[19] L. Zhang, W. Chen, Y. Hu and C. C. Chen, “Statistical Static Timing Analysis with Conditional Linear MAX/MIN Approximation and Extended Canonical Timing Model,” *IEEE Transactions on Computer-Aided Design*, vol. 25, no. 6, pp. 1183-1191, June 2006.

[20] A. Ramalingam, A. K. Singh, S. R. Nassif, G.-J. Nam, M. Orshansky and D. Z. Pan, “An Accurate Sparse Matrix Based Framework for Statistical Static Timing Analysis,” *IEEE/ACM Int’l Conf. Computer Aided Design*, San Jose, CA, Nov.

2006.

[21] J. Wang, Z. Yue, X. Lu, W. Qiu, W. Shi and D. M. H. Walker, “A Vector-based Approach for Power Supply Noise Analysis in Test Compaction,” *IEEE Int’l Test Conf.*, Austin, TX, Nov. 2005.

[22] Y. M. Jiang, A. Krstic, and K. T. Cheng, “Dynamic Timing Analysis Considering Power Supply Noise Effects,” *Int’l Symp. on Quality of Electronic Design*, San Jose, CA, March 2000, pp. 137-143.

[23] J. J. Liou, A. Krstic, Y. M. Jiang and K. T. Cheng, “Path Selection and Pattern Generation for Dynamic Timing Analysis Considering Power Supply Noise Effects,” *IEEE/ACM Int’l Conf. Computer Aided Design*, San Jose, CA, Nov. 2000, pp. 493-497.

[24] A. Krstic and K. T. Cheng, “Vector Generation for Maximum Instantaneous Current Through Supply Lines for CMOS Circuits,” *ACM/IEEE Design Automation Conf.*, Anaheim, CA, June 1997, pp. 383-388.

[25] G. Bai, S. Bodda and I. N. Hajj, “Static Timing Analysis Including Power Supply Noise Effect on Propagation Delay in VLSI Circuits,” *ACM/IEEE Design Automation Conf.*, Las Vegas, NV, Jun. 2001, pp. 295-300.

[26] D. Kouroussis and F. N. Najm, “A Static Pattern-Independent Technique for Power Grid Voltage Integrity Verification,” *ACM/IEEE Design Automation Conf.*, Anaheim, CA, June, 2003, pp. 99-104.

[27] H. Kriplani, F. Najm and I. Hajj, “Pattern Independent Maximum Current Estimation in Power and Ground Buses of CMOS VLSI Circuits: Algorithms

Signal Correlations, and Their Resolution," *IEEE Transactions on Computer-Aided Design*, vol. 14, no. 8, pp. 998-1012, August 1995.

[28] D. F. Morrison, Multivariate Statistical Methods. New York: McGraw-Hill, 1976.

[29] W. Qiu, J. Wang, D. M. H. Walker, D. Reddy, X. Lu, Z. Li, W. Shi and H. Balachandran, "K Longest Paths Per Gate (KLPG) Test Generation for Scan-Based Sequential Circuits," *IEEE Int'l Test Conf.*, Charlotte, NC, Oct. 2004, pp. 223-231.

[30] M. Hashimoto, J. Yamaguchi and H. Onodera, "Timing Analysis Considering Spatial Power/Ground Level Variation," *IEEE/ACM Int'l Conf. Computer Aided Design*, San Jose, CA, Nov. 2004, pp. 814-820.

[31] H. Chang and S.S. Sapatnekar, "Statistical Timing Analysis Under Spatial Correlations," *IEEE Transactions on Computer-Aided Design*, vol. 24, no. 9, pp. 1467-1482, Sep. 2005.

VITA

Hyun Sung Kim received his Bachelor of Science degree in Computer Science and Engineering from The University of California at Davis in June 2003. He came to Texas A&M University in September 2003 and graduated in August 2007 with a Master of Science in Computer Engineering. His research interests are statistical static timing analysis with voltage noise variations and electronic design automation of VLSI circuits. He can be reached at 4027 Country Club Dr. #201, Los Angeles, CA 90019. His email address is howdybill@gmail.com.