# Probabilistic Analysis of Interconnect Coupling Noise

Sarma Vrudhula, *Member, IEEE*, David T. Blaauw, *Member, IEEE*, and Supamas Sirichotiyakul

***Abstract*—Noise simulators and noise avoidance tools are playing an increasingly critical role in the design of deep submicron circuits. However, noise estimates produced by these simulators are often very pessimistic. For large, high-performance industrial ICs, which can contain hundreds of thousands of nets, the worst case estimates of the noise results in thousands of reported violations, without any information about the likelihood of the possible noise violation. In this paper, we present a probabilistic approach to prioritize the violating nets based on the likelihood of occurrence of the reported noise. We derive an upper bound on the probability that the total noise injected on a given victim net by a specific set of aggressors exceeds a threshold. This is equivalent to a lower bound on the expected number of clock cycles required to realize the noise violation for the first time, i.e., mean time-to-failure. If the probability of a failure in a victim is sufficiently small, it is possible that even during the operation of the part for a number of years, the probability of failure on the net is negligible and the net can be assigned a lower priority for the application of noise avoidance strategies. We demonstrate the utility of this approach through experiments carried out on an large industrial processor design using a state-of-the-art industrial noise analysis tool.**

***Index Terms*—Capacitive coupling, crosstalk noise, functional noise, interconnect analysis, noise estimation.**

## I. Introduction

NOISE DUE to capacitive coupling of interconnects has become an important reliability issue in the design of deep submicron (DSM) circuits. Advances in process technology have reduced the wire widths and wire spacing, resulting in taller, thinner, and closer wires [1]. The cross-coupling capacitance between wires has become a dominant component of the total wire capacitance [1]–[4], and can cause a degradation in performance or a malfunction of the circuit. A glitch can occur on an otherwise stable net (victim) due to switching of its neighboring nets (aggressors) and can then propagate to a storage element or a dynamic node, which could alter the circuit state. Noise causing this type of failure is referred to as *functional noise*. Alternatively, if the victim net is also switching, then the noise injected onto the victim by the aggressors can result in a significant change in the delay [2], thereby impacting the clock frequency of the circuit. This type of noise is referred to as *delay noise*. For static circuits, functional noise may not be as severe as delay noise. However, the increasing use of dynamic circuits makes consideration of functional noise equally important. In addition, the likelihood of either type of malfunction has increased significantly due to lower supply voltages and higher clock frequencies. Thus, methods for accurate estimation of functional and delay noise and techniques for reducing their effects have become critical in the design of DSM circuits.

Key features of noise simulators [5], [6] include: 1) models of the network consisting of the victim and its aggressors, as well as models of the gates that drive them; 2) methods for estimating the resistance and capacitance of each net and the coupling capacitance to neighboring nets; 3) criteria for filtering out insignificant aggressors of each net; 4) methods for computing the noise threshold at the input of the victim's receiver gate; and 5) a simulation engine that numerically solves the network equations to compute the voltage waveform of the noise induced on the victim.

As with any aspect of circuit design, there is a tradeoff between the accuracy of the noise estimation techniques and the time required to compute them. Fast techniques are essential since many high-performance industrial circuits can have hundreds of thousands of nets. Consequently, most often each *cluster* consisting of a victim and its associated set of aggressors, is modeled as a linear network, so that the composite noise waveform due to all of the aggressors can be computed as the sum of the noise waveforms due to each aggressor.

Even with linear models of the nonlinear drivers [6], [7], the time required for SPICE-level simulation of a fully distributed *RC* network is unacceptably high, and hence, reduced order models [8], [9] are used in noise simulators. For constructing noise avoidance strategies or for delay analysis in the presence of interconnect coupling, further simplifications are necessary. For example, the coupling capacitance ($C_c$) can be replaced with an equivalent grounded capacitance in the range of $3 * [0, C_c]$ in static timing analysis [10]. Closed-form expressions for the noise waveform have been developed under a variety of simplifying assumptions, and have been applied to noise avoidance using driver sizing, wire spacing, and wire sizing [11]–[16].

Regardless of the models and the approximations used, noise analysis tools are designed to be pessimistic due to the consequences of missing a potential malfunction. However, in large industrial designs, this has resulted in thousands of reported failures, which are expensive to fix in terms of chip area, chip performance, and design time. The noise injected onto a victim by an aggressor during a clock cycle $k$ is a time-varying function that depends on the pair of input vectors applied at cycles

Manuscript received July 25, 2001; revised January 28, 2002 and July 30, 2002. This work was supported in part by the National Science Foundation Center for Low Power Electronics under Grant EEC-9523338 and in part by the National Science Foundation under Grant CCR-0205227. This paper was recommended by Associate Editor M. Pedram.

S. Vrudhula was with the Advanced Design Tools Group, Motorola, Inc., Austin, TX 78729 USA. He is now with the Electrical and Computer Engineering Department, University of Arizona, Tucson, AZ 85721 USA (e-mail: sarma@ece.arizona.edu).

D. T. Blaauw was with the Advanced Design Tools Group, Motorola, Inc., Austin, TX 78729 USA. He is now with the Electrical Engineering and Computer Science Department, University of Michigan, Ann Arbor, MI 48109-2122 USA (e-mail: blaauw@umich.edu).

S. Sirichotiyakul was with the Advanced Design Tools Group, Motorola, Inc., Austin, TX 78729 USA. She is now with Sun Microsystems, Burlington, MA 01803 USA.

Digital Object Identifier 10.1109/TCAD.2003.816212

$k-1$ and $k$, the transition times of the signals and the time at which signal transitions occur. In the case of delay noise, the switching and transition times of the aggressor depend not only on its gate and wiring delays, but also on the waveform at the victim's output [17]. The root cause of the pessimistic estimates of noise is the way in which the noise waveforms from each aggressor net are combined to form the composite noise waveform. A common approach [5]–[7] is to assume that all aggressor nets can switch in the same direction and at the same time, and that their peaks will be aligned. This results in a *worst case* maximum height composite noise waveform. Often, however, the number of aggressors with significant amount of coupled noise can be large, exceeding five to ten aggressors. In such a case, the likelihood of realizing a worst case composite noise, where all aggressors are required to switch at exactly the right time, is small.

A number of deterministic methods have been proposed to more accurately estimate the noise waveform by accounting for the logic [18]–[20] and timing relationships [5], [21]–[24] between the aggressors and a victim. One approach is to perform static timing analysis (ignoring the functional information) and construct *timing windows* which specify the earliest and latest time that a signal can switch within a clock cycle. The peaks of the two aggressors would be aligned only if their timing windows overlap [6]. For delay noise, determining the alignment of the aggressors and relating that to the maximum delay is more involved. In [22], an convergent iterative algorithm is presented that accounts for the fact that the aggressor timing window depends on the victim's output.

Noise estimates can be improved as well by taking into account logic correlations [6], [23]–[26]. For example, when considering functional noise, even if the timing windows of two aggressors overlap, can be determined that the two signals never switch in the same direction, then the noise value is taken as the maximum of the respective peak values instead of the sum of their peak values. In [26], a precise characterization of such signal interactions based entirely on the logic or functional characteristics is given. This allows identification of signals that can interfere as input to noise avoidance tools. In [23], the problem of identifying the vector pair that would result in two nets switching within a given interval is formulated as a constrained optimization of a pseudo-Boolean function.

Although the incorporation of timing information in noise analysis improves the accuracy of the noise estimates, typically, the timing windows obtained from static timing analysis are wide and eliminate only a small portion of the reported failures. In addition, for large circuits, only a limited amount of logic correlations can be derived because physical proximity of nets and logic dependencies between them need not be related. Moreover, these methods require high computational effort. Even after accounting for logic and timing correlations, noise reports for high-performance circuits with hundreds of thousands of nets often identify thousands of nets with potential violations, and provide only the worst case noise associated with each victim. Deterministic methods, therefore, can not completely address the problem inherent in the pessimistic assumptions of coupled noise analysis.

In this paper, a method for estimating the *likelihood* that a total noise (*functional*) on a victim net will exceed a given threshold is presented. If the noise on a victim net originates from a significant number of aggressors, the likelihood of all aggressors combining in a worst case manner is small. If this likelihood is sufficiently small, it is possible that even during operation of the part for a number of years, the probability of a failure (i.e., a noise violation) on the net is negligible, and the net can be assigned a lower priority for the application of noise-avoidance strategies or even eliminated from further consideration. The probability of a failure on a victim net will decrease as the number of aggressor nets involved increases and as their timing windows become wider. One simple criterion to prioritize nets is the expected number of clock cycles before the first violation occurs on a specific victim. We refer to this quantity as the net's *mean time-to-failure* (MTF).

It is important to note that there is a basic distinction between the approach taken here and the classical statistical approach that might be used to predict noise. In the latter, the set of all nets is viewed as an ensemble and noise associated with a net is viewed as a random variable over the ensemble. Such an approach would not lead to accurate statements about any specific net. The approach described here characterizes a victim by its set of aggressors, each of which contributes a specific noise waveform (a deterministic function of time obtained from simulation) to the victim. The *randomness* or variability of the noise on a victim net is over different clock cycles and arises due to the times when each aggressor switches over a specified time interval within a clock period.

The rest of the paper is organized as follows. Section II contains a description of the models. A discussion of the assumptions appears in Section III. Sections IV and V contain the main technical contribution, namely, the upper bounds on the probability that the noise on a victim will exceed a given threshold, and estimates of its MTF. Results of experiments carried out on a large, high-performance PowerPC microprocessor are presented in Section VI. A brief description of a recently developed industrial noise simulator called ClariNet [6], [7], which was used in the analysis is also given in this section. Section VII contains a summary.

## II. Probabilistic Model of Noise

In the remainder of the paper, we refer to a victim with a set of $n$ aggressors as a *cluster of size* $n$. Since we are examining functional noise, without loss of generality, we assume that the victim is stable at logic 0, and an aggressor switches from logic 0 to logic 1.

To accommodate a very large number of nets, noise simulators use linear models for the victim and aggressor driver gates and construct a distributed *RC* network for each aggressor–victim combination. The composite noise waveform is obtained by taking the sum of the noise waveforms resulting from each aggressor. From linear circuit theory, the general form of the noise waveform seen on the victim due to an aggressor switching is a sum of weighted, decaying exponentials, the number of terms being equal to the order of the circuit. To simplify the algebraic work, we take a linear approximation for the *rising* and *falling* portions of the waveform. Consequently, the noise waveform resulting from each aggressor is approximated by a triangular pulse. In addition, we associate a *timing window* $[a_i, b_i]$ with aggressor $i$, where $a_i$ and $b_i$ denote the earliest and latest possible arrival times of the transition. Thus, the noise waveform resulting from aggressor $i$, which is denoted by $Z_i(t)$, is represented by $(h_i, r_i, d_i, a_i, b_i)$, where $h_i$ is the peak noise voltage and $r_i$ and $d_i$ are the slopes of the rising and falling edges of the noise waveform, respectively.

As mentioned in Section I, the randomness or variability is over different clock cycles, and is a result of the random point in time when each aggressor switches. Let $\tau_i$ be the random variable that denotes the time instance in $[a_i, b_i]$ at which aggressor $i$ switches. We assume that $\tau_i$ is uniformly distributed over $[a_i, b_i]$, i.e.,

$$F_{\tau_i}(x) = \text{Prob}(\tau_i \leq x) = \frac{(x-a_i)_+ - (x-b_i)_+}{b_i - a_i} \quad (1)$$

where $(x)_+$ denotes the *ramp* function which is defined as $(x)_+ = x$ if $x \geq 0$, otherwise $(x)_+ = 0$. This assumption is not very restrictive and the approach can be extended to other distributions of switching events. However, in the absence of any other information, the uniform distribution is the meaningful choice. For a given value of $\tau_i$, $Z_i(t)$ is expressed as

$$Z_i(t) = \begin{cases} r_i(t-\tau_i), & \tau_i \leq t \leq \frac{h_i}{r_i} \\ d_i(\frac{h_i}{r_i} - (t-\tau_i)) + h_i, & \frac{h_i}{r_i} \leq t \leq \frac{h_i}{r_i} + \frac{h_i}{d_i} + \tau_i \\ 0, & \text{elsewhere} \end{cases} . \quad (2)$$

With $\tau_i$ being a random variable, $Z_i(t)$ is a stochastic process, and for a fixed value of $t$, $Z_i(t)$ is a random variable. Let $F_{Z_i(t)}(z) = \text{Prob}(Z_i(t) \leq z)$ denote the distribution function of $Z_i(t)$. For a fixed value of $t$, if $N$ waveforms given by (2) are generated corresponding to $N$ sample observations of $\tau_i$, then $F_{Z_i(t)}(z)$ represents the fraction of those $N$ waveforms that at time $t$ have a value $\leq z$. For a fixed $t$

$$\{Z_i(t) \leq z\} = \left\{\tau_i \geq t - \frac{z}{r_i}\right\} \cup \left\{\tau_i \leq t - \frac{h_i}{r_i} - \frac{h_i}{d_i} + \frac{z}{r_i}\right\}. \quad (3)$$

This leads directly to the distribution function of $Z_i(t)$, expressed in terms of the distribution function of $\tau_i$.

$$F_{Z_i}(z) = \begin{cases} 1 - F_{\tau_i}\left(t - \frac{z}{r_i}\right) + F_{\tau_i}\left(t - \frac{h_i}{r_i} - \frac{h_i}{d_i} + \frac{z}{d_i}\right), & 0 \leq z \leq h_i \\ 1, & z \geq h_i \\ 0, & \text{otherwise} \end{cases} . \quad (4)$$

Let $S_n(t)$ represent the waveform of the total noise on a given victim net in a cluster of size $n$. As stated earlier, the use of linear models for the victim and aggressor drivers, allows us to represent the total noise on a victim as a sum of the noise due to each aggressor. We assume that the random variables $Z_i(t)$, $i = 1, 2, \ldots, n$, are independent. To account for the fact that an aggressor may or may not switch within a clock period, we introduce a binary random variable $X_i$ associated with aggressor $i$, where $X_i = 1$ with probability $p_i$ and $X_i = 0$ with probability $(1-p_i)$. $p_i$ is called the switching probability of aggressor $i$. The random variable that represents the total noise on the victim is

$$S_n(t) = Z_1(t) * X_1 + Z_2(t) * X_2 + \cdots + Z_n(t) * X_n. \quad (5)$$

The process $S_n(t)$ is not stationary and, therefore, its statistics depend on $t$. Furthermore, the noise waveforms resulting from the individual aggressors often differ widely in scale and, therefore, the assumption that they are random samples from the same population cannot be justified. As a result, it is not possible to derive a closed-form expression for the probability distribution of $S_n(t)$. The alternative, which is to carry out numerical convolution for each $t$, would be computationally prohibitive. Therefore, we proceed with the next best alternative, which is to derive bounds on the tail probability of $S_n(t)$. The bounds that we derive are based on first obtaining expressions for all of the moments of $S_n(t)$. Before we proceed with this, we examine the basic assumptions made in the model.

## III. Discussion of Assumptions

The objective of this paper is to provide a method to estimate the likelihood of a (functional) noise violation reported by a noise simulator that assumes a worst case scenario. An analytical approach is needed for this, since Monte Carlo simulation would be prohibitively expensive given the large number of nets that have to be processed in a industrial setting. Consequently, several simplifying assumptions were made to arrive at such a solution. These are as follows.

1) The time at which each aggressor switches is uniformly distributed within its timing interval.
2) The noise pulse of a given aggressor is approximated by a triangular shaped pulse.
3) The individual aggressor signals are independent.

The first assumption can be relaxed and the analysis extended to other distributions. It was made primarily because existing timing verifiers only provide an interval and no distribution. However, the results can be extended to other distributions. An example would be a triangular distribution [27], which can be intuitively justified.

The second assumption seems to be necessary for analytical tractability. The noise pulse due to an aggressor is more accurately represented by a sum of weighted exponentials. However, this would not lead to an analytical solution. In the interest of being conservative when constructing the bound on the expected number of clock cycles for the first violation, it is a simple to construct a triangular pulse that approximates a given exponential pulse.

The third assumption, namely, independence of the aggressors, is technically an orthogonal issue under a zero-delay model. That is, under the zero delay model, the determination of logic correlations (which is an intractable problem in itself) is done separately and only those aggressors that could be switching in the same direction would be included in a cluster. These have been accounted for in the experiments. However, temporal correlations are far more difficult to model and no effective solution to include them and still maintain analytical tractability, is known. It should be pointed out that if a group of nets is highly correlated (logically or temporally), then the probabilistic analysis may be optimistic. This would be the case when, for example, a signal crosses a bus, and all lines of the bus switch at the same time. One approach to address this problem is to identify nets that are correlated and treat them as a single net in the probabilistic analysis, i.e., with a single timing window and a peak noise height equal to the sum of the noise heights of each line of the bus.

## IV. Criterion for Prioritizing Nets

A simple criterion that designers can use to prioritize nets that are identified as having potential noise violations is the *expected* number of clock cycles before the first violation occurs. We refer to this as the MTF. A noise violation is said to occur on a victim net if the total noise $S_n(t)$ on that victim exceeds a given threshold $\alpha$. The threshold is calculated based on the characteristics of the victim's receiver gate. On each clock cycle, we observe a realization of $S_n(t)$. Let $N(t)$ be the random variable that denotes the number of clock cycles required to observe the first noise violation *at time* $\mathrm{t}$. Assuming independence of noise waveforms over different clock cycles, the probability that the noise on this victim will exceed $\alpha$ for the first time on $k$th cycle is given by [28]

$$\mathrm{Prob}\big(\mathrm{N(t)=k}\big) = [P(S_n(t)\le\alpha)]^{k-1} * \Big(1 - P\big(S_n(t)\le\alpha\big)\Big). \tag{6}$$

Let $\mathcal{E}$ denote the expectation operator. The average or expected number of clock cycles before the first violation occurs at time $t$ within a clock period, is given by

$$\mathrm{MTF}(t) = \mathcal{E}\big(N(t)\big) = \frac{1}{P\big(S_n(t)>\alpha\big)}. \tag{7}$$

When assigning a victim net a very low priority or even discarding it from further consideration because its $\mathrm{MTF}(t)$ is very large (e.g., five years), we should ensure that $t$ is selected so that, at no other value of $t$, the $\mathrm{MTF}(t)$ is smaller. Thus, we have

$$t^* = \{t \mid \mathrm{MTF}(t) \text{ is minimum}\} \tag{8}$$

$$\mathrm{MTF}^* = \mathrm{MTF}(t^*). \tag{9}$$

To compute $\mathrm{MTF(t)}$, we need the distribution function of $S_n(t)$. An analytical form does not appear to be possible. The alternative, which is to carry out numerical convolution for each $t$, would be computationally prohibitive. Therefore, we proceed with the next best alternative, which is to derive bounds on $P(S_n(t)>\alpha)$.

## V. Bounds on Noise Probability

A common strategy to construct an upper bound on the probability that a nonnegative random variable exceeds a given value is to construct a parametric family of upper bounds and then find the value of the parameter that minimizes the upper bound. This approach is based on the Chernoff bound [29], which states that

$$P\big(S_n(t)>\alpha\big) \le e^{-\theta\alpha}\Phi_{S_n(t)}(\theta)\ \forall\theta\ge 0 \tag{10}$$

where $\Phi_{S_n(t)}(\theta)$ is the moment-generating function (mgf) of $S_n(t)$, and $\theta$ is an unknown parameter to be determined. From (5) and from the properties of the mgf, we have

$$\Phi_{S_n(t)}(\theta) = \prod_{i=1}^{n}\big(p_i\Phi_{Z_i(t)}(\theta) + 1 - p_i\big). \tag{11}$$

By definition of the mgf, we have

$$\Phi_{Z_i(t)}(\theta) = \sum_{k=0}^{\infty}\frac{\mathcal{E}\big(Z_i^k(t)\big)\theta^k}{k!} \tag{12}$$

where $\mathcal{E}(Z_i^k(t))$ is the expectation of the $Z_i^k(t)$, which is the $k$th order moment of $Z_i(t)$.

Let $B(\theta,t,\alpha) = e^{-\theta\alpha}\Phi_{S_n(t)}(\theta)$. The value of $\theta$ that minimizes $B(\theta,t,\alpha)$ is the solution to (13) [29]

$$\frac{\mathcal{E}\big(S_n(t)e^{\theta S_n(t)}\big)}{\mathcal{E}\big(e^{\theta S_n(t)}\big)} = \frac{\Phi'_{S_n(t)}(\theta)}{\Phi_{S_n(t)}(\theta)} = \alpha. \tag{13}$$

Equation (13) can be simply expressed as

$$\frac{d\log\big(\Phi_{S_n(t)}(\theta)\big)}{d\theta} = \alpha. \tag{14}$$

Using (11) and (14), the value of $\theta$ that minimizes the bound given by the right-hand side of (10) is the solution to the equation

$$\sum_{i=1}^{n}\frac{p_i\Phi'_{Z_i(t)}(\theta)}{p_i\Phi_{Z_i(t)}(\theta)+1-p_i} = \alpha. \tag{15}$$

We have now established a method to compute a lower bound on MTF. First, we have to determine $t^*$ [see (8)]. Then we have to minimize $B(\theta,t^*,\alpha)$ with respect to $\theta$. Before we proceed with this task, we need expressions for $\mathcal{E}(Z_i^k(t))$ [see (11) and (12)].

### A. Moments of Total Noise

By definition, $\mathcal{E}(Z_i^k(t)) = \int_{-\infty}^{\infty} z^k dF_{Z_i(t)}(z)$ and $F_{Z_i(t)}(z)$ is given by (1). Integrating by parts and noting that $F_{Z_i(t)}(z)$ has a finite jump at $z=0$, we obtain the following recurrence relation:

$$\mathcal{E}\big(Z_i^k(t)\big) = h_i^k - k\int_0^{h} z^{k-1}F_{Z_i(t)}(z)dz. \tag{16}$$

Equation (16) can be solved exactly. The derivation appears in Appendix A. The exact form is given by

$$\begin{aligned}
&\mathcal{E}\big(Z_i^k(t)\big)\\
&\quad = 0^k F_{Z_i(t)}(0) + \frac{1}{(b_i-a_i)(k+1)!r_i}\\
&\qquad \cdot\Bigg[\left(h_i + r_i\left(t-b_i-\frac{h_i}{r_i}\right)_+\right)^{k+1}\\
&\qquad\quad - \left(h_i + r_i\left(t-a_i-\frac{h_i}{r_i}\right)_+\right)^{k+1}\\
&\qquad\quad + \left(r_i(t-a_i)_+\right)^{k+1} - \left(r_i(t-b_i)_+\right)^{k+1}\Bigg]\\
&\qquad + \frac{1}{(b_i-a_i)(k+1)!d_i}\\
&\qquad \cdot\Bigg[\left(h_i - d_i\left(t-b_i-\frac{h_i}{r_i}\right)_+\right)^{k+1}\\
&\qquad\quad - \left(h_i - d_i\left(t-a_i-\frac{h_i}{r_i}\right)_+\right)^{k+1}\\
&\qquad\quad + \left(-d_i\left(t-a_i-\frac{h_i}{r_i}-\frac{h_i}{d_i}\right)_+\right)^{k+1}\\
&\qquad\quad - \left(-d_i(t-b_i-\frac{h_i}{r_i}-\frac{h_i}{d_i})_+\right)^{k+1}\Bigg]
\end{aligned} \tag{17}$$

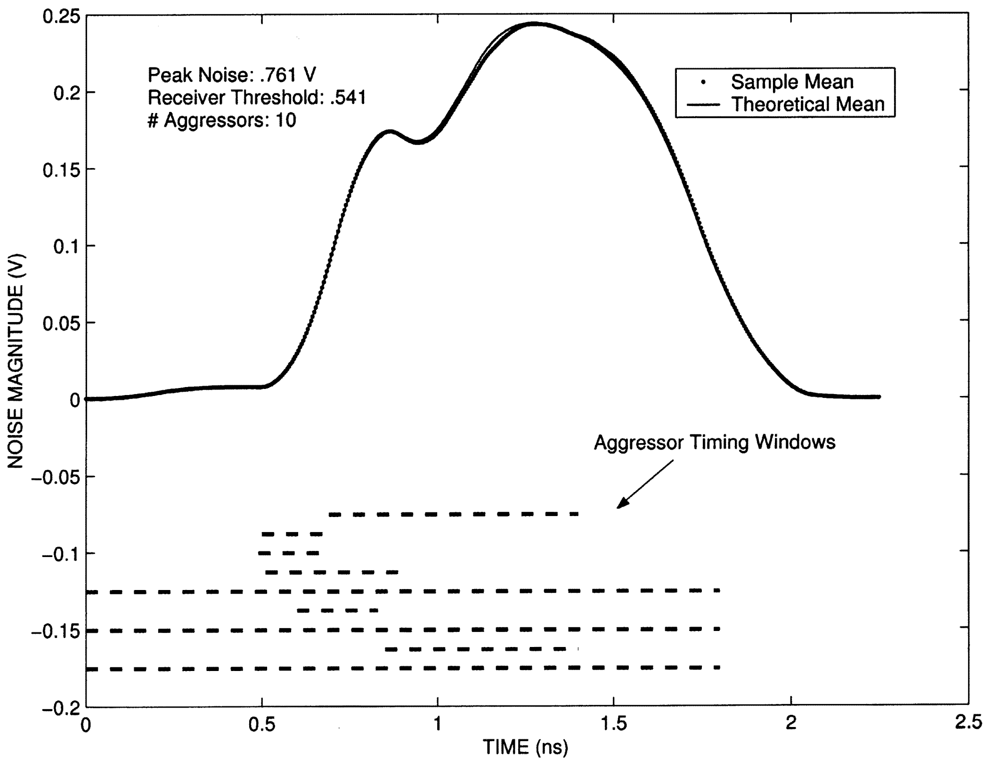


Fig. 1. Theoretical and sample mean of total noise of one victim with ten aggressors.

where $F_{Z_i(t)}(0)$ [using (4) and (1)] is given by

$$\begin{aligned} F_{Z_i(t)}(0) =& 1 - \frac{1}{b_i - a_i}\left[(t-a_i)_+ - (t-b_i)_+\right] \\ &+ \frac{1}{b_i - a_i}\left[\left(t - \left(\frac{h_i}{r_i} + \frac{h_i}{d_i} + a_i\right)\right)_+ \right. \\ &\left. - \left(t - \left(\frac{h_i}{r_i} + \frac{h_i}{d_i} + b_i\right)\right)_+\right]. \end{aligned} \tag{18}$$

Equation (17) expresses the $k$th order moment of the noise waveform due to aggressor $i$, in terms of its descriptors, $(h_i, r_i, d_i, a_i, b_i)$. The moments of $S_n(t)$ can be obtained from (17) and (5). To see how the theoretical and sample moments compare, the first four moments were computed for many clusters taken from a PowerPC microprocessor. Using the timing intervals for each aggressor that were obtained from static timing analysis, and the noise estimates produced by the simulator [6], a MonteCarlo simulation was carried out by varying the switching point of each aggressor. For each selection of switching points, the composite waveform was computed and was repeated 5000 times. This corresponds to 5000 clock cycles. Figs. 1–4 show plots of the theoretical [using (17)] and sample mean, standard deviation, and the third and fourth moments. The timing intervals associated with each aggressor are shown at the bottom of each plot. As can be seen from these plots, there is very good agreement between the theoretical and sample moments.

Since the ordinary moments involve only simple powers of $t$, the mgf $\Phi_{Z_i(t)}(\theta)$ can be obtained almost by inspection. Using the definition of $\Phi_{Z_i(t)}(\theta)$ given in (12) and substituting the expression for $\mathcal{E}(Z_i^k(t))$ given in (17) into (12), we obtain

$$\begin{aligned} &\Phi_{Z_i(t)}(\theta) \\ &\quad = F_{Z_i(t)}(0) + \frac{1}{(b_i - a_i) r_i \theta} \\ &\qquad \cdot \left[\exp\left(\theta\left(h_i + r_i\left(t - b_i - \frac{h_i}{r_i}\right)_+\right)\right)\right. \\ &\qquad - \exp\left(\theta\left(h_i + r_i\left(t - a_i - \frac{h_i}{r_i}\right)_+\right)\right) \\ &\qquad \left. + \exp(\theta r_i (t-a_i)_+) - \exp(\theta r_i (t - b_i)_+\right] \\ &\qquad + \frac{1}{(b_i - a_i) d_i \theta} \\ &\qquad \cdot \left[\exp\left(\theta\left(h_i + d_i\left(t - b_i - \frac{h_i}{r_i}\right)_+\right)\right)\right. \\ &\qquad - \exp\left(\theta\left(h_i + d_i\left(t - a_i - \frac{h_i}{r_i}\right)_+\right)\right) \\ &\qquad + \exp\left(-\theta d_i\left(t - \left(\frac{h_i}{r_i} + \frac{h_i}{d_i} + a_i\right)_+\right)\right) \\ &\qquad \left. - \exp\left(-\theta d_i\left(t - \left(\frac{h_i}{r_i} + \frac{h_i}{d_i} + b_i\right)\right)_+\right)\right]. \end{aligned} \tag{19}$$

$\Phi_{S_n(t)}(\theta)$ [the mgf of $S_n(t)$] is computed by substituting (19) into (11).

Before we proceed with the optimization of the bound given in (10), it will be instructive to see what the bounds look like as a function of $\theta$ and $t$. Figs. 5 and 6 show plots of the bound in

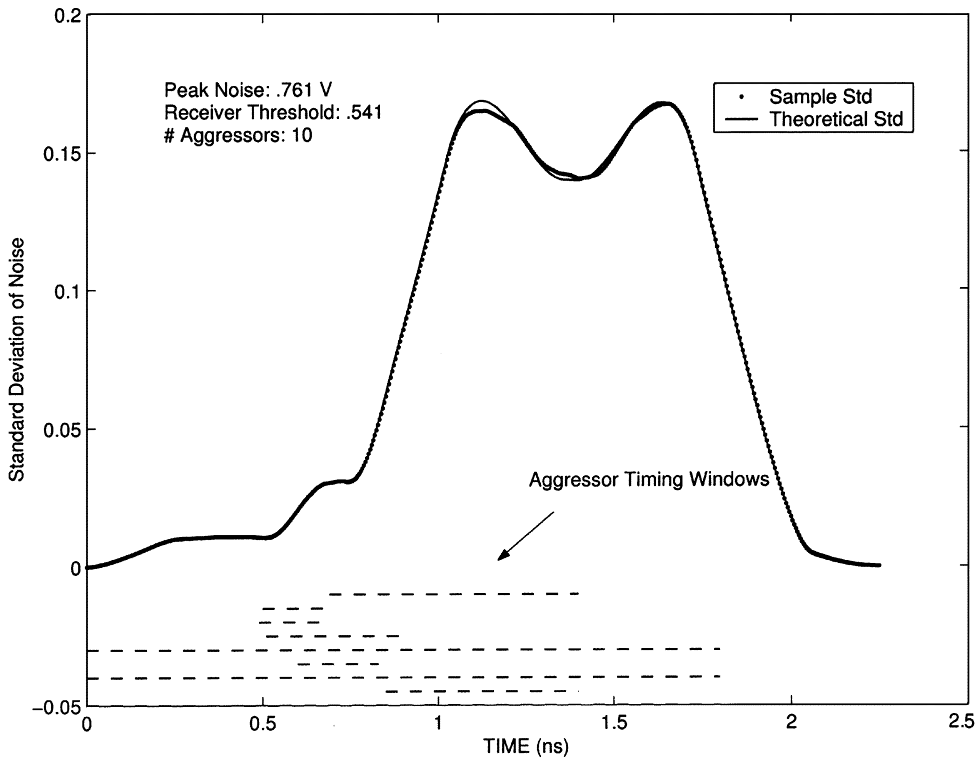


Fig. 2. Theoretical and sample standard of total noise of one victim with ten aggressors.

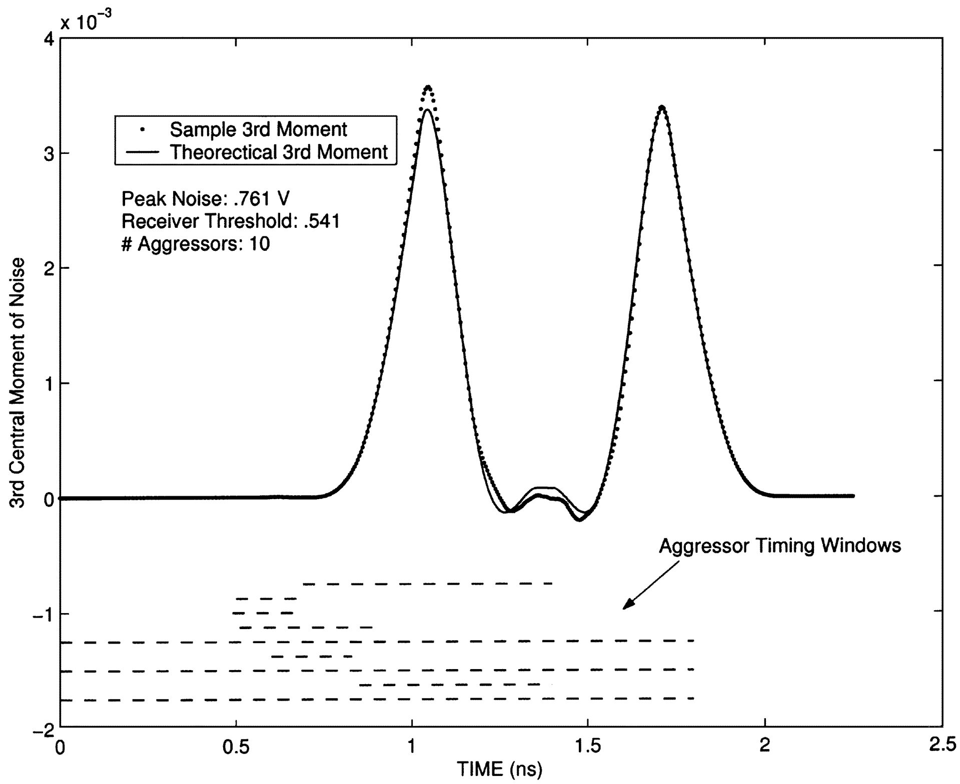


Fig. 3. Theoretical and sample third moment of total noise of one victim with ten aggressors.

(10) for a cluster of size ten, without and with timing intervals, respectively. Note that ignoring the timing intervals of aggressors simply means that the timing intervals span the entire clock period. In the time interval of interest, Figs. 7 and 8 show that the bound (10) as a function of $\theta$ is a convex surface. Figs. 9 and 10 show that once the desired value of $t$ is obtained, then the optimal value of $\theta$ can be obtained very quickly by either gradient or direct search techniques.

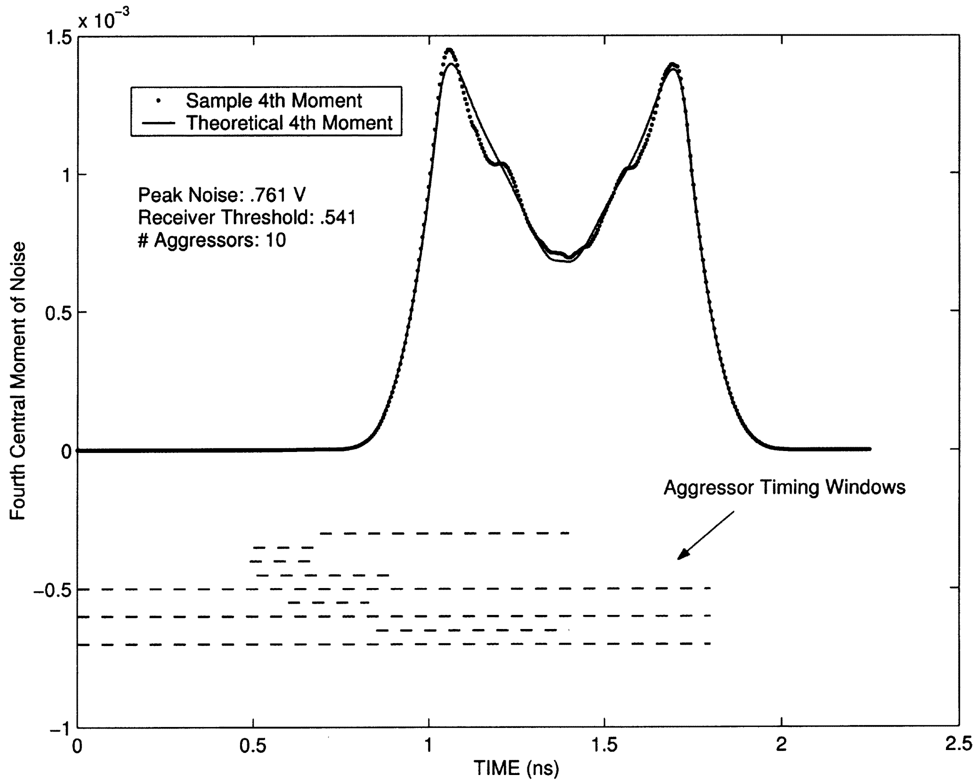


Fig. 4. Theoretical and sample fourth moment of total noise of one victim with ten aggressors.

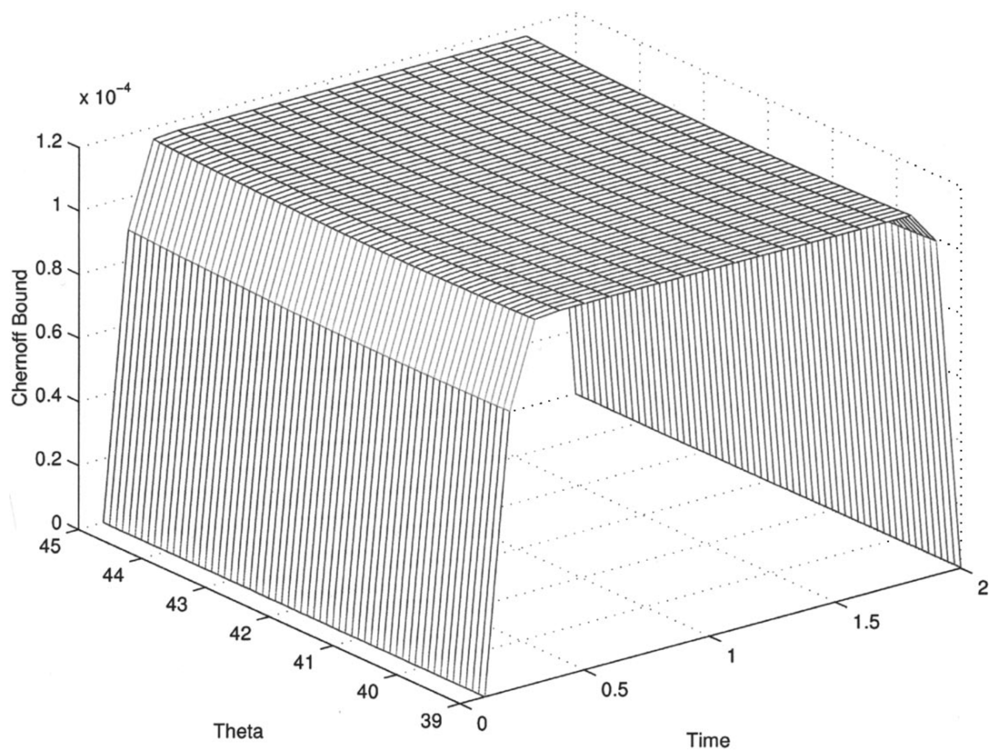


Fig. 5. Chernoff bound of a net with ten aggressors without timing intervals.

## B. Method for Computing the Bound

In Section IV, we stated that a lower bound on the expected number of clock cycles to observe the first violation (total noise exceeding a given threshold) requires determining a value of $t$ at which $\mathrm{MTF}(t)$ is minimum, or equivalently the value of $t$ that maximizes $B(\theta, t, \alpha)$. That value of $t$ is denoted by $t^*$. Since $\theta$ is also unknown, an iterative search along $\theta$ and $t$ would be prohibitively time consuming, given the large number of clusters that have to be processed. Moreover, convergence is not guaran-

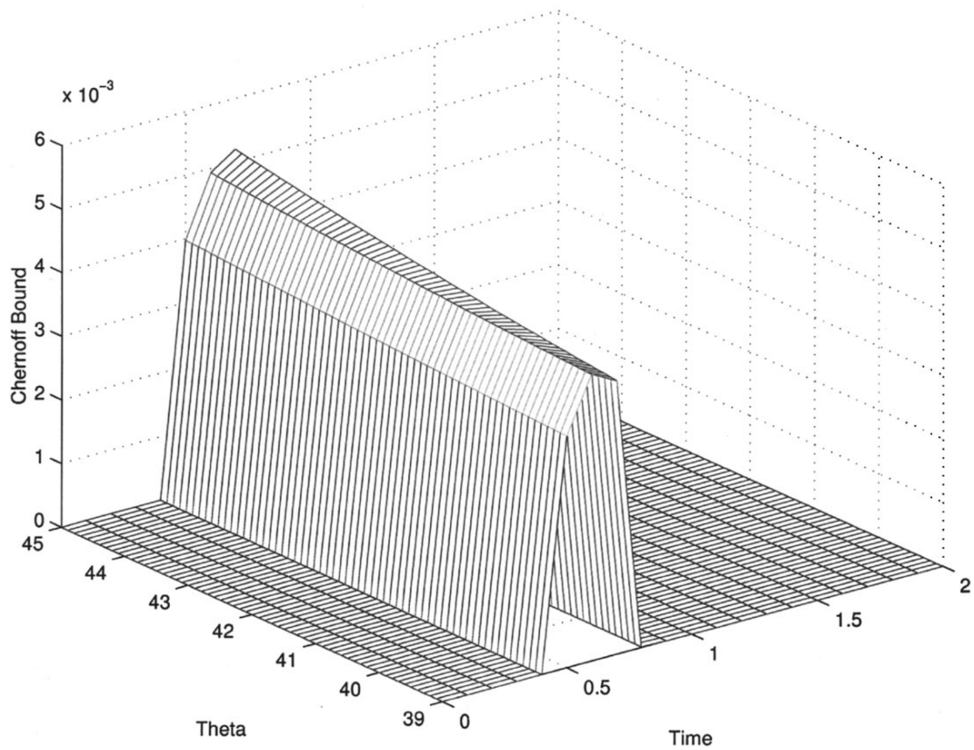


Fig. 6. Chernoff bound of a net with ten aggressors with timing intervals.

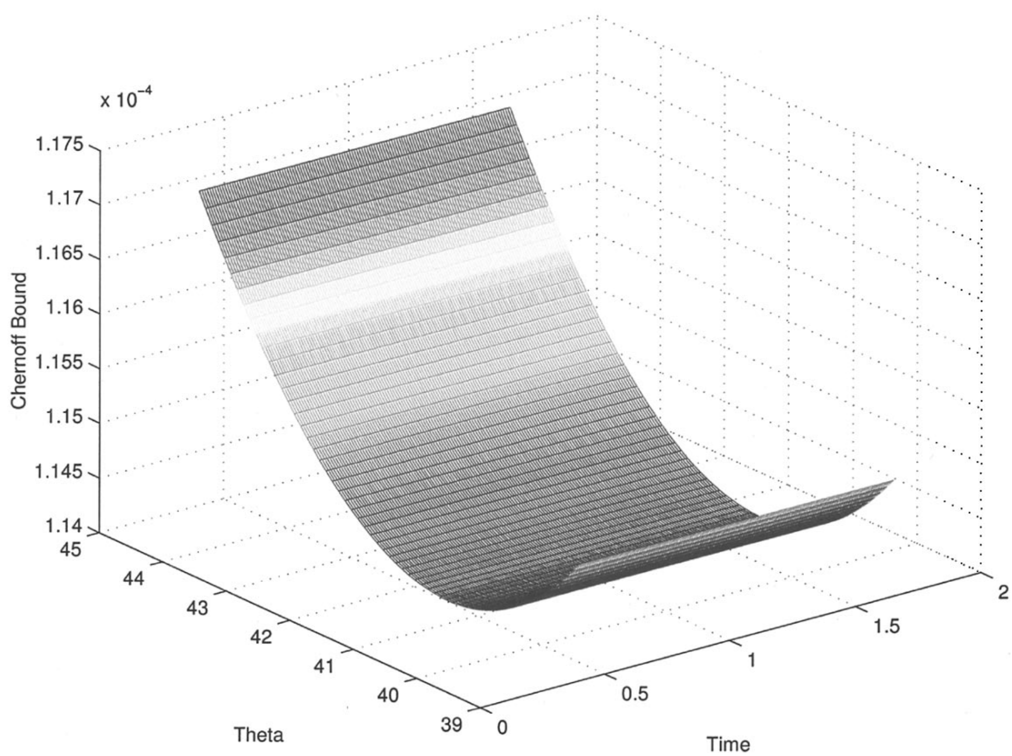


Fig. 7. Plot in Fig. 5 restricted to smaller range of $t$.

teed. We now describe a novel procedure that identifies $t^*$ first, without having to know $\theta$, and then the bound $B(\theta, t*, \alpha)$ can be easily minimized with respect to $\theta$ using gradient search.

The solution to finding $t^*$ is based on the monotonicity properties of the mgf $\Phi_{Z_i(t)}(\theta)$. These properties allow us to identify a *finite* set ($4n$ for a cluster of size $n$) of distinguished time points $\mathcal{T}$, which we refer to as *breakpoints*. Now for a fixed $\theta$, the maximum value of $\Phi_{S_n(t)}(\theta)$ will occur at the breakpoints or in between two breakpoints. The latter may occur because at some breakpoint, the mgf of some of the aggressors are increasing, while the mgf of others are decreasing. In such a situation, we would have to iteratively search, between every pair

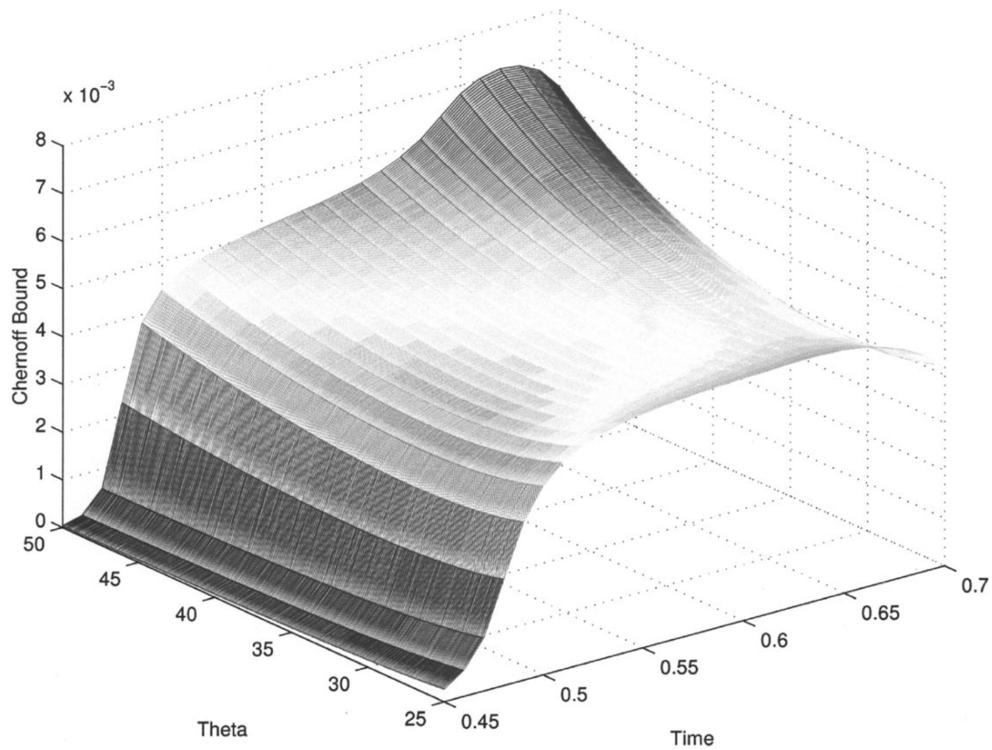


Fig. 8. Plot in Fig. 6 restricted to smaller range of $t$.

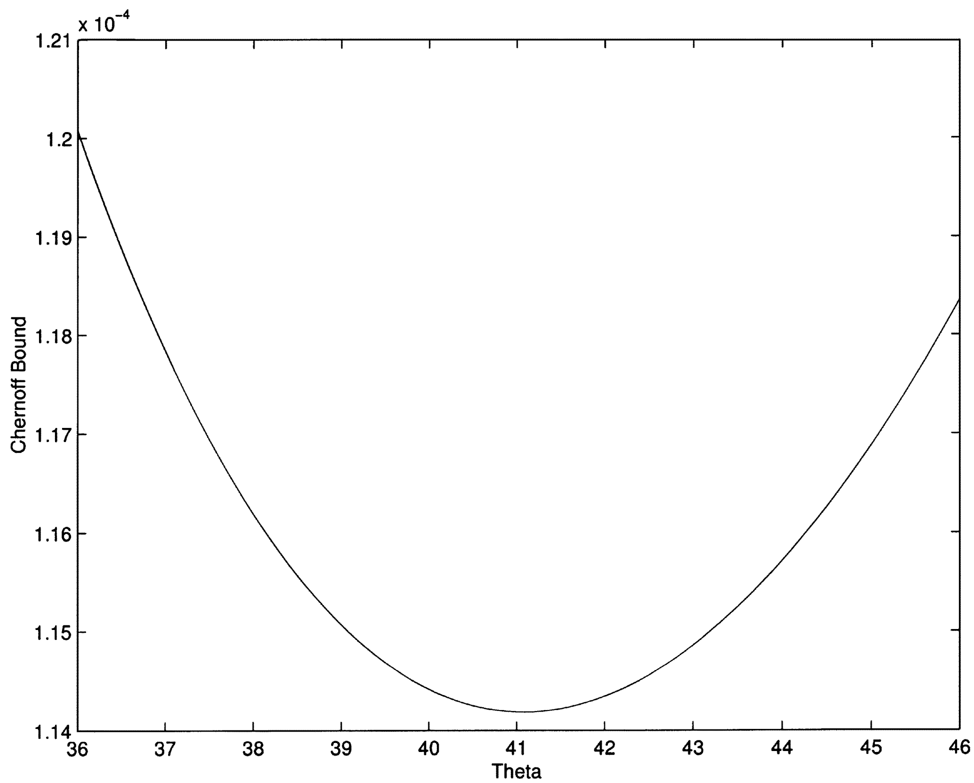


Fig. 9. Plot in Fig. 5 for a single value of $t$.

of such breakpoints, for the value of $t$, where $\Phi_{S_n(t)}(\theta)$ attains a maximum. Moreover, the point within such an interval where the maximum of $\Phi_{S_n(t)}(\theta)$ occurs will generally depend on $\theta$. To avoid this, we construct a modified mgf $\tilde{\Phi}_{S_n(t)}(\theta)$ such that

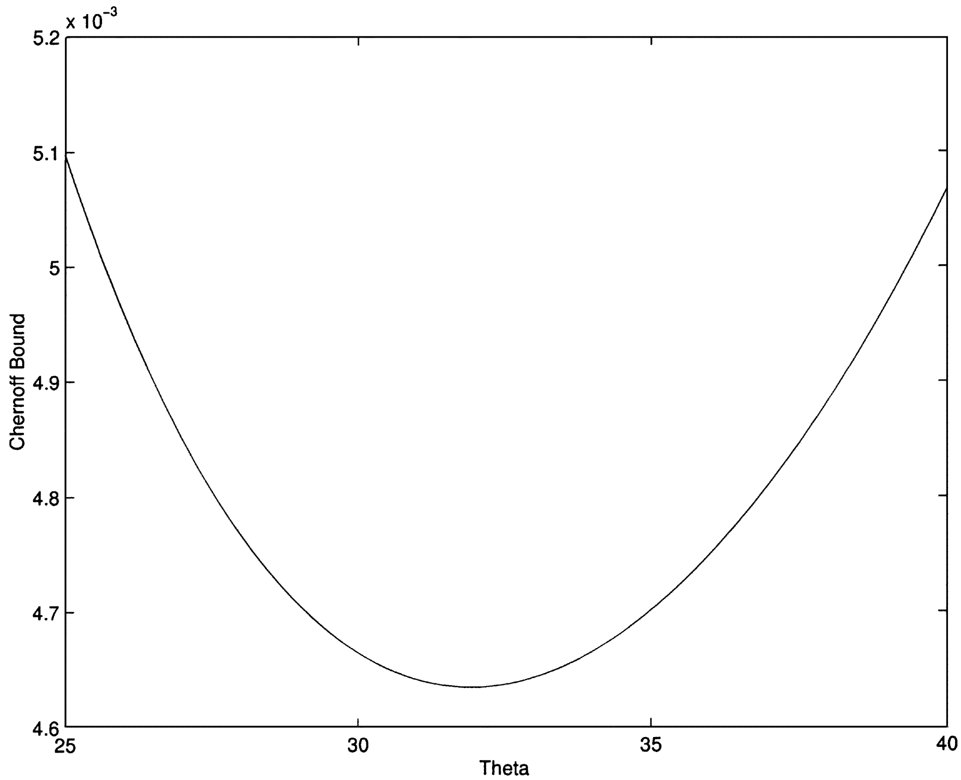


Fig. 10. Plot in Fig. 6 for a single value of $t$.

$\tilde{\Phi}_{S_n(t)}(\theta) \geq \Phi_{S_n(t)}(\theta)$, for all $\theta$, and so that only the breakpoints in $\mathcal{T}$ need to be examined in order to determine the value of $t$, where $\tilde{\Phi}_{S_n(t)}(\theta)$ is maximum. Furthermore, many of the points in $\mathcal{T}$ can be discarded. Finally, at each time point in the reduced set of breakpoints, we solve (15) to find $\theta$ using gradient search and choose that value of $\theta$ that minimizes the bound. Note that this is a conservative solution in that it will lead to a smaller lower bound on MTF*.

Consider the mgf of $Z_i(t)$ given in (19). For a fixed $\theta$, its behavior as a function of $t$ can be classified into two cases.

*Case 1)* $b_i - a_i \geq h_i/r_i + h_i/d_i$*:* This situation occurs when the width of the timing interval is greater than the width of the noise pulse. In this case, $\Phi_{Z_i(t)}(\theta)$ is:

- monotonically increasing for $t \in [a_i, h_i/r_i + h_i/d_i + a_i)$;
- constant for $t \in [h_i/r_i + h_i/d_i + a_i, b_i)$;
- monotonically decreasing for $t \in [b_i, h_i/r_i + h_i/d_i + b_i]$.

*Case 2)* $b_i - a_i < h_i/r_i + h_i/d_i$*:* In this case, $\Phi_{Z_i(t)}(\theta)$ is:

- monotonically increasing for $t \in [a_i, \tilde{t}_i]$;
- monotonically decreasing for $t \in (\tilde{t}_i, h_i/r_i + h_i/d_i + b_i]$

where $\tilde{t}_i$ is given by

$$\tilde{t}_i = \frac{d_i}{r_i + d_i}\left(\frac{h_i}{r_i} + \frac{h_i}{d_i} + a_i\right) + \frac{r_i}{r_i + d_i} b_i. \tag{20}$$

The points where $\Phi_{Z_i(t)}(\theta)$ changes direction are the *breakpoints* of $Z_i(t)$. The breakpoints of all of the aggressors are collected into a list $\mathcal{T}$ and sorted in increasing order. Formally, $\mathcal{T} = \cup_i \tau_i$, where $\tau_i$ is defined as

$$\tau_i = \begin{cases} \left\{a_i, \frac{h_i}{r_i} + \frac{h_i}{d_i} + a_i, b_i, \frac{h_i}{r_i} + \frac{h_i}{d_i} + b_i\right\}, & \text{if case 1} \\ \left\{a_i, \tilde{t}_i, \frac{h_i}{r_i} + \frac{h_i}{d_i} + b_i\right\}, & \text{otherwise} \end{cases}.$$

With $n$ aggressors, the maximum possible number of points in $\mathcal{T}$ is $4n$. However, most of the points in $\mathcal{T}$ can be eliminated from further consideration. This is done by associating a direction $\delta_{i,j}$ with $\Phi_{Z_i(t_j)}(\theta)$ for each $t_j \in \mathcal{T}$.

$$\delta_{i,j} = \begin{cases} -1, & \text{if } \Phi_{Z_i(t_j)}(\theta) \text{ is decreasing} \\ 0, & \text{if } \Phi_{Z_i(t_j)}(\theta) \text{ is constant} \\ 1, & \text{if } \Phi_{Z_i(t_j)}(\theta) \text{ is increasing} \end{cases}.$$

Now $\tilde{\Phi}_{Z_i(t)}(\theta)$ can be constructed from $\Phi_{Z_i(t)}(\theta)$ as follows. Let $\mathcal{T} = \{t_1, t_2, \ldots, t_m\}$.

$$\tilde{\Phi}_{Z_i(t_1)}(\theta) = \Phi_{Z_i(t_1)}(\theta)$$

$$\tilde{\Phi}_{Z_i(t_j)}(\theta) = \begin{cases} \Phi_{Z_i(t_{j-1})}(\theta), & \text{if } \delta_{i,j-1} = -1 \text{ and } \exists k, \delta_{k,j-1} = 1 \\ \Phi_{Z_i(t_j)}(\theta), & \text{otherwise} \end{cases}.$$

and $\tilde{\Phi}_{S_n(t)}(\theta) = \prod_{i=1}^{n} \tilde{\Phi}_{Z_i(t)}(\theta)$.

As stated above, many of the points in $\mathcal{T}$ can be discarded. Let $t_j$ be the first point in $\mathcal{T}$ such that $\delta_{i,j} = -1$, for some $i$. Let $t_k$ be the last point in $\mathcal{T}$ such that $\delta_{i,k} = 1$, for some $i$. Then points in $(t_1, \ldots, t_{j-1})$ and $(t_{k+2}, \ldots, t_m)$ can be discarded. This is because $t_j$ and $t_k$ are the first and last points where one of the modified mgfs $\tilde{\Phi}_{Z_i(t)}(\theta)$ has reached its peak. Note that this reduction is possible only with $\tilde{\Phi}_{Z_i(t)}(\theta)$ and not with $\Phi_{Z_i(t)}(\theta)$. Once a time point $t^* \in \mathcal{T}$, where $\tilde{\Phi}_{S_n(t)}(\theta)$ is maximum is identified, then the value of $\theta$ that minimizes $\tilde{B}(\theta, t^*, \alpha)$ is computed by numerically solving (15) with $\tilde{\Phi}$ replacing $\Phi$.

## VI. EXPERIMENTAL RESULTS

The noise simulator that was used is called ClariNet [6], which is an industrial noise analysis tool that was developed

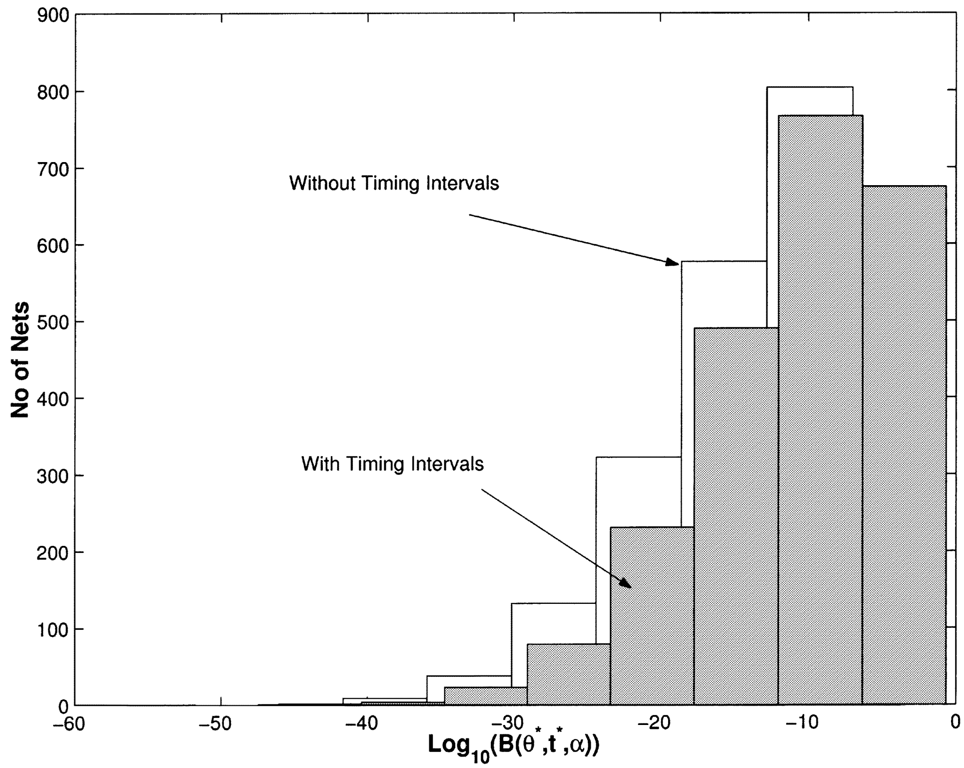


Fig. 11. Histogram of the $\log_{10}(B(\theta^*, t^*, \alpha))$.

to analyze large, high-performance processor designs. The simulator embodies several features that help speed up noise analysis, allowing it to process hundreds of thousands of nets in a few hours.

The noise simulator iterates over all nets to analyze their noise. First, the victim net is reduced to a simplified network which is guaranteed to overestimate the noise. The calculated noise is compared against a designer-specified acceptable noise value and if it is smaller, the victim net passes the noise analysis. Three filters with increasing complexity are used sequentially to quickly eliminate those nets that are guaranteed to pass noise analysis. If the net does not pass the noise filters, we linearize the aggressor and driver gates. The noise on the victim net is calculated using linear superposition where the noise induced by each aggressor is simulated while grounding the other aggressor voltage sources. We then use the logic and timing correlations to determine the subset of aggressors that induce the maximum possible noise. The combined noise from these aggressors is added to the propagated noise from the previous stage which is a predetermined noise threshold voltage. This aggregate noise pulse is propagated through the victim receiver gate by simulating it. If the noise peak at the output of the receiver gate is greater than the predetermined noise threshold, a noise failure is reported. For faster execution, the aggregate noise peak can be compared against a precharacterized table of ac noise. Further details of this tool are available in [6] and [7].

The experimental results reported in this paper were generated by performing noise analysis for a high-performance PowerPC microprocessor. The total number of nets analyzed was nearly 200 000. The total number of nets reported as having a noise violation was 2141. Each net was analyzed twice—once for *low overshoot* (victim net is to be stable at logic 0 and its aggressors switch from logic 0 to logic 1) and once for *high undershoot* (victim net is stable at logic 1 and aggressors switch from logic 1 to logic 0). As a result, the total number of violations was 2501. The maximum cluster size (number of aggressors per victim) net was set to ten. For each net, the noise report indicates the peak height and width of the noise injected on the net by each aggressor, and the threshold of the receiver gate. This is data that was used for the probabilistic analysis as described in this paper.

For each cluster, the optimal value of the bound was computed, with $\alpha$ set to the receiver's threshold. Fig. 11 shows a histogram of the $\log_{(10)}(B(\theta^*, t^*, \alpha))$. Note that a value less than $-16.94$ on the abscissa of the histogram represents an MTF of more than five years for a continuously running 555-MHz processor. For this particular experiment, 634 out of 2501, or 25.35% of the violations were in this category. Fig. 12 shows a plot of the percentage of nets that have an MTF greater than or equal to the value on the abscissa. It is important to note that the set of all nets, when using timing intervals (lower plot in Fig. 12), is much smaller than the set of nets when timing intervals are ignored.

The bounds on the MTF depend on the switching probabilities for each aggressor [see (11)]. The computation of the switching probability of each net is a very complex problem and various methods have been reported in the literature, especially in works addressing dynamic power estimation. The problem requires examining both temporal and spatial correlations of signals [30]. Computing the switching probability is itself a difficult problem, and the existing techniques are computationally expensive. Moreover, the number of nets we have to process is

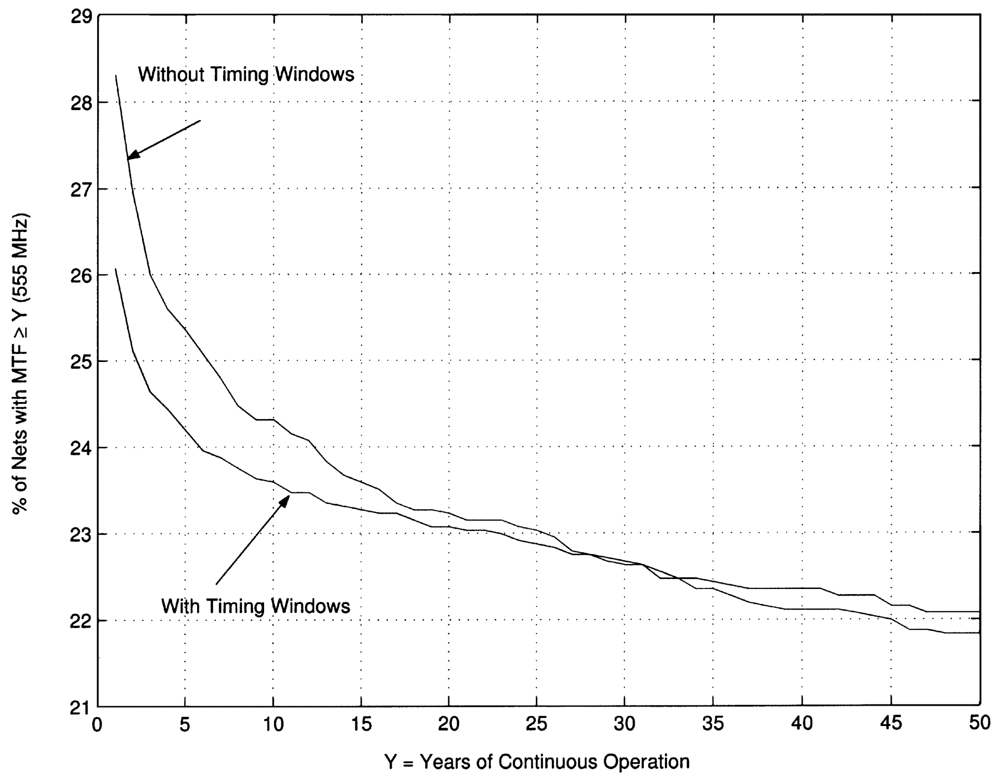


Fig. 12. MTF of nets ($\text{switching prob.} = 0.2$).

on the order of hundreds of thousands. Hence, in the experiments we performed, the switching probability of each net was set to 0.2, which is the value used by the designers to estimate the average dynamic power consumption. Fig. 13 shows a plot of the MTF versus years for three different switching probabilities. As expected, the most pessimistic results would be for the case where every net is switched. The percentage of nets with an MTF of $\geq 50$ years is approximately, 22%, 13%, and 8%, corresponding to switching probabilities of 0.2, 0.5, and 1.0.

We now examine the MTF of the whole chip. Suppose only those nets whose MTF is less than $y$ years are *fixed*, (i.e., modified in some way to reduce or eliminate the noise). Let $N(y)$ denote the set of nets whose MTF is greater than or equal to $y$ years. If a noise violation occurs on any net in $N(y)$, then the chip is deemed to have failed. Assuming noise violations on different nets are independent events, the probability that the chip will fail due to the nets in $N(y)$ is given by

$$\text{Prob Chip Fails} = \mathrm{PCF}\big(N(y)\big) = 1 - \prod_{i \in N(y)} (1 - p_i) \quad (21)$$

where $p_i$ is the probability that there is a noise violation on net $i$, for $i \in N(y)$. The Chernoff bound on $p_i$ is used to obtain an upper bound on PCF and a corresponding lower bound on the MTF of the chip.

We are interested in determining the maximum number of nets in $N(y)$ that can be ignored and still have an MTF of the chip to be greater than or equal to $y$ years. That is, once $N(y)$ is determined, we determine the largest subset $M(y) \subseteq N(y)$ such that $\mathrm{PCF}(N(y)) \leq yC_y$, where $C_y$ is the number of clock cycles per year. Fig. 14 shows a plot of $|M(y)|$ versus $y$. From this figure, we see that the number of nets that we can possibly ignore based on the chip MTF $\geq 5$ years is 422 (16.87% of the nets). This is in contrast to the 634 (25.35% of the nets) whose MTF individually is $\geq 5$ years. The plot also indicates that the number of nets that can be ignored which a chip MTF of $\geq 50$ years is 13.63%.

Next, we examined how the ranking changed when it is based on the MTF versus when it is based on the peak noise. We deleted the nets whose MTF was greater than or equal to five years. The number of nets remaining was 1867. For these nets, we computed the MTF* values. This provides an alternative ranking based on the expected number of clock cycles required to observe a noise violation for the first time, rather than simply the sum of all of the noise peaks injected by each aggressor. Two sorted lists $L_{\mathrm{PEAK}}$ and $L_{\mathrm{MTF}}$ of the nets were generated. $L_{\mathrm{PEAK}}$ is the set of nets in decreasing order of the magnitude of the peak noise of the composite waveform. $L_{\mathrm{MTF}}$ is the set of nets in increasing order of their MTF* values. Fig. 15 shows a plot of the percentage of nets in the original list $L_{\mathrm{PEAK}}$ that retain their ranking in the second list $L_{\mathrm{MTF}}$, as each list is traversed. The abscissa represents the top $X\%$ of the nets. The ordinate represents the percentage of the nets in the top $X\%$ of list $L_{\mathrm{PEAK}}$ that remained in the top $X\%$ of the nets in list $L_{\mathrm{MTF}}$. For example, from the top 20% of the list $L_{\mathrm{PEAK}}$ which represents 373 nets, only 12.5% of *them* ($\approx 46$ nets) remained in the top 20% of $L_{\mathrm{MTF}}$. If the ranking of the nets did not change, then the plot would be a horizontal line at 1. The important conclusion here that the probabilistic approach identifies nets based not only on the noise magnitude, but also on the likelihood of occurrence. Nets for which the simulator reports a relatively small

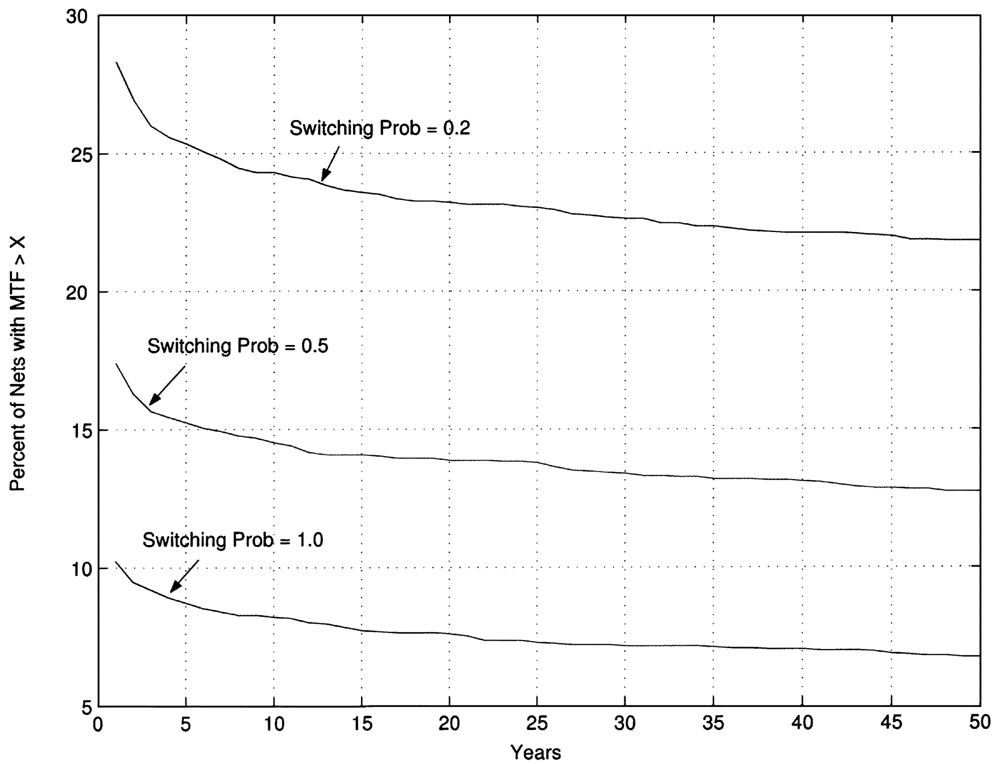


Fig. 13. MTF versus years for different switching probabilities.

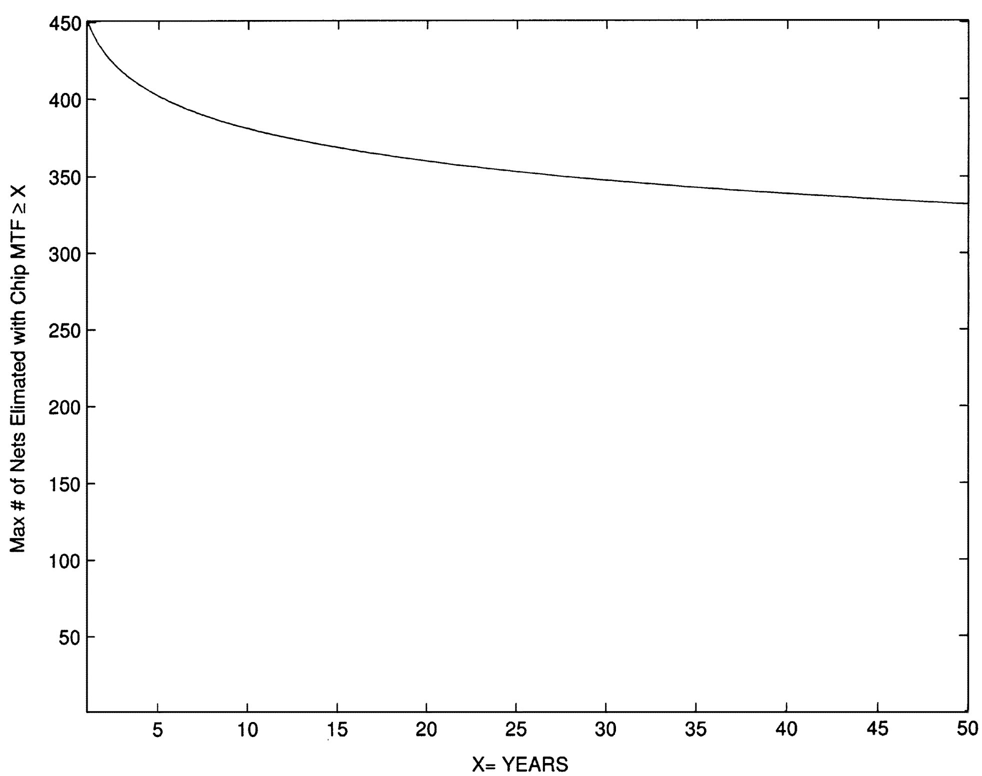


Fig. 14. Number of nets versus chip MTF.

noise violation may become more important, resulting in corrective action being taken on them before nets with larger noise magnitudes.

The next generation of the processor chip was also simulated. In this run, no limit was set on the size of a cluster, i.e., on the maximum number of aggressors associated with a victim. In this case, the total number of noise violations reported by the noise simulator was 429. This run would include clusters with a large number of small aggressors. Table I shows the result of this experiment. It is seen that the percentage of nets with MTF greater

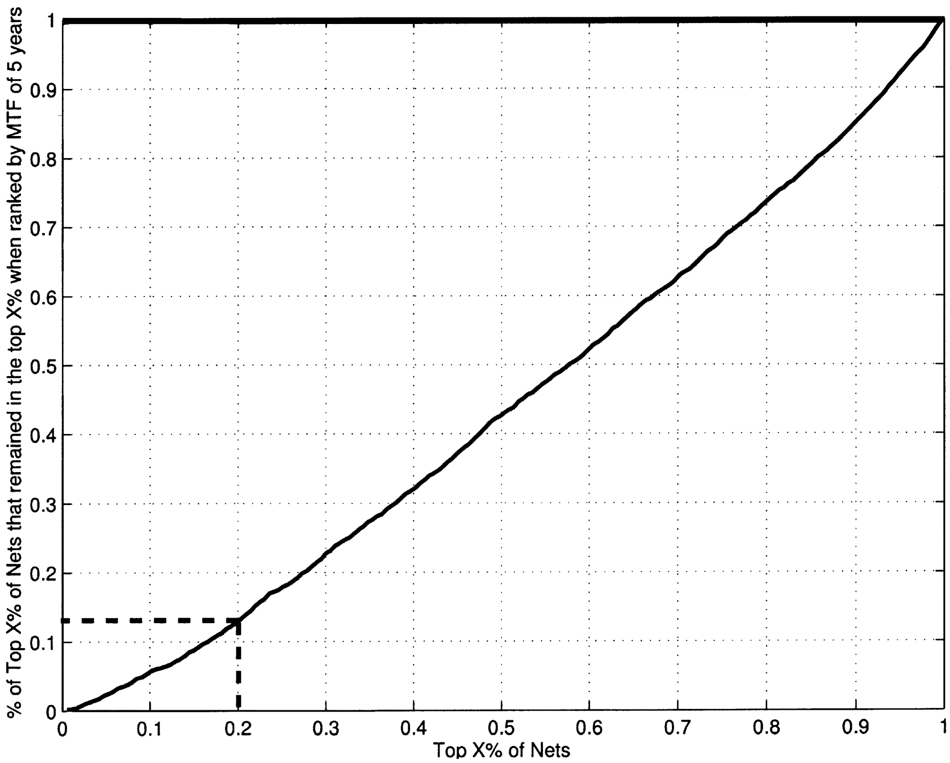


Fig. 15. Results of ranking of nets by likely noise magnitude.

TABLE I
PERCENT OF NETS WITH MTF > Y YEARS VERSUS Y FOR NEXT GENERATION CHIP

| Y (years) | % of Nets with MTF > Y |
|---|---|
| 1 | 41.96 |
| 2 | 41.26 |
| 3 | 41.03 |
| 4 | 41.03 |
| 5 | 40.79 |
| 10 | 40.33 |
| 20 | 40.33 |
| 30 | 40.09 |
| 40 | 40.09 |
| 50 | 39.36 |

than a given number of years is larger than in the previous run. This is to be expected, as the likelihood of a noise violation decreases when there are a larger number of small aggressors.

## VII. CONCLUDING REMARKS

In this paper, we presented a new approach toward analyzing functional noise due to capacitive coupling of interconnects. The primary objective was to provide a measure of the likelihood of a noise violation to rank the nets. This could be used to select nets for possible application of noise avoidance strategies. Several simplifying assumptions were made (e.g., concerning the nature of the noise pulse due to each aggressor, the distribution of their switching times, and logic and temporal correlations among different aggressors) to arrive at a analytical solution. Even with the restrictions, only upper bounds on the likelihood of a noise violation or equivalently, a lower bound on the expected number of clock cycles before the first violation were obtained. The method was exercised on an industrial high-performance microprocessor, using a recently developed state-of-the-art noise simulator. There are several ways this work can be extended and improved. The most obvious is to extend the analysis to other, possibly more realistic distributions of aggressor switching times. We believe that temporal correlations should be used as a preprocessing step (e.g., group highly correlated aggressors into a single aggressor, etc.), since attempting to include them in the analysis significantly increases the complexity of the analysis. The most useful direction to extend this approach is toward the analysis of delay noise [27].

## APPENDIX

### A. Solution to Equation 16

$$\begin{aligned}\mathcal{E}(Z_i^k(t)) = &\int_{-\infty}^{0} z^k dF_{Z_i(t)}(z) + \int_0^h z^k dF_{Z_i(t)}(z) \\ &+ \int_h^{\infty} z^k dF_{Z_i(t)}(z)\end{aligned} \tag{22}$$

$$\begin{aligned}\int_{-\infty}^{0} z^k dF_{Z_i(t)}(z) =& 0^k \left[F_{Z_i(t)}(0) - F_{Z_i(t)}(0^-)\right] \\ =& 0^k F_{Z_i(t)}(0)\end{aligned} \tag{23}$$

$$\begin{aligned}\int_0^h z^k dF_{Z_i(t)}(z) =& h^k F_{Z_i(t)}(h) - 0^k F_{Z_i(t)}(0) - \\ & k\int_0^h z^{k-1} F_{Z_i(t)}(z) dz\end{aligned} \tag{24}$$

$$\int_h^\infty z^k dF_{Z_i(t)}(z) = h^k \left[F_{Z_i(t)}(h^+) - F_{Z_i(t)}(h)\right] = 0 \quad (25)$$

$$\begin{aligned}\mathcal{E}(Z_i^k(t)) &= h^k F_{Z_i(t)}(h) - k\int_0^h z^{k-1} F_{Z_i(t)}(z)dz \\ &= h^k - k\int_0^h z^{k-1} F_{Z_i(t)}(z)dz. \end{aligned} \quad (26)$$

For a fixed $t$, let $V_0(z) = F_{Z_i(t)}(z)$, $V_j(z) = \int V_{j-1}(z)dx$, and consider the integral

$$I_k = k\int_0^h z^{k-1} F_{Z_i(t)}(z)dz.$$

Integrating by parts $k$ times, we obtain, (with $k_{(j)} = k(k-1)\cdots(k-j+1)$),

$$I_k = \sum_{j=1}^k (-1)^{j-1} k_{(j)} \left[h_i^{k-j} V_j(h_i) - 0^{k-j} V_j(0)\right]. \quad (27)$$

Substituting the above into the RHS of (16) and noting that $V_0(h_i) = 1$ and $V_0(0) = F_{Z_i(t)}(0)$, we obtain

$$\begin{aligned}&\mathcal{E}(Z_i^k(t)) \\ &= \sum_{j=0}^k [(-1)^j k_{(j)} h_i^{k-j} V_j(h_i) - (-1)^k k! V_k(0) + 0^k F_{Z_i(t)}(0)].\end{aligned} \quad (28)$$

$V_j(z)$ is obtained by using the definitions of $F_{Z_i(t)}(z)$ and $F_{\tau_i}(t)$ given in (4) and (1). In particular, $V_j(h_i)$ and $V_j(0)$ can be expressed as

$$\begin{aligned} V_j(h_i) =& \frac{h_i^j}{j!} + \frac{d_i^j - (-r_i)^j}{(b_i - a_i)(j+1)!} \\ &\cdot \left[\left(t - a_i - \frac{h_i}{r_i}\right)_+^{j+1} \right. \\ &\left. - \left(t - b_i - \frac{h_i}{r_i}\right)_+^{j+1}\right] \end{aligned} \quad (29)$$

$$\begin{aligned} V_k(0) =& \frac{0^k}{k!} + \frac{(-1)^{k+1} r_i^k}{(b_i - a_i)(k+1)!} \\ &\cdot \left[(t-a_i)_+^{k+1} - (t-b_i)_+^{k+1}\right] \\ &+ \frac{d_i^k}{(b_i - a_i)(k+1)!} \\ &\cdot \left[\left(t - a_i - \frac{h_i}{r_i} - \frac{h_i}{d_i}\right)_+^{k+1} \right. \\ &\left. - \left(t - b_i - \frac{h_i}{r_i} - \frac{h_i}{d_i}\right)_+^{k+1}\right]. \end{aligned} \quad (30)$$

Substituting (29) and (30) into (28), carrying out the summation term by term, and after considerable simplification, the result expressed in (17) is obtained.

## REFERENCES


[1] D. Sylvester and K. Keutzer, "Getting to the bottom of deep submicron," in *Proc. Int. Conf. Computer-Aided Design*, Nov. 1998, pp. 203–211.

[2] Y. Cao, T. Sato, X. Huang, C. Hu, and D. Sylvester, "New approaches to noise-aware static timing analysis," in *Proc. ACM/IEEE Int. Workshop Timing Issues Specification Synthesis Digital Syst.*, Dec. 2000, pp. 8–13.

[3] A. Devgan, "Efficient coupled noise estimation for on-chip interconnects," in *Proc. Int. Conf. Computer-Aided Design*, Nov. 1997, pp. 147–151.

[4] K. L. Sheppard and V. Narayanan, "Noise in deep submicron digital design," in *Proc. Int. Conf. Computer-Aided Design*, Nov. 1996, pp. 524–531.

[5] K. L. Sheppard, V. Narayanan, P. C. Elementary, and G. Zheng, "Global harmony: Coupled noise analysis for full-chip RC interconnect networks," in *Proc. Int. Conf. Computer-Aided Design*, Nov. 1997, pp. 139–146.

[6] R. Levy, D. Blaauw, G. Braca, A. Dasgupta, A. Grinshpon, C. Oh, B. orshav, S. Sirichotiyakul, and V. Zoltov, "ClariNet: A noise analysis tool for deep submicron design," in *Proc. Design Automation Conf.*, June 2000, pp. 223–238.

[7] S. Sirichotiyakul, D. Blaauw, R. Levy, C. Oh, S. Shams, V. Zolotov, and J. Zuo, "Driver modeling and alignment of worst-case delay noise," in *Proc. ACM/IEEE Int. Workshop Timing Issues Specification Synthesis Digital Syst.*, Dec. 2000, pp. 1–7.

[8] A. Odabaioglu, M. Celik, and L. T. Pileggi, "PRIMA: Passive reduced-order interconnect macromodeling algorithm," in *Proc. Int. Conf. Computer-Aided Design*, Nov. 1997, pp. 139–146.

[9] L. T. Pillage and R. A. Rohrer, "Asymptotic waveform analysis for timing analysis," *IEEE Trans. Computer-Aided Design*, pp. 352–366, Apr. 1990.

[10] A. B. Kahng, S. Muddu, and E. Sarto, "Switch factor based analysis of coupled RC interconnects," in *Proc. Int. ASIC/SoC Conf.*, 1999, pp. 3–8.

[11] J. Cong, D. Z. Pan, and P. V. Srinivas, "Improved crosstalk modeling for noise constrained interconnect optimization," in *Proc. ACM/IEEE Int. Workshop Timing Issues Specification Synthesis Digital Syst.*, Dec. 2000, pp. 14–20.

[12] C. J. Alpert, A. Devgan, and S. T. Quay, "Buffer insertion for noise and delay optimization," in *Proc. Design Automation Conf.*, 1998, pp. 362–367.

[13] M. R. Becer, D. Blaauw, R. Levy, S. Sirichotiyakul, C. Oh, V. Zolotov, J. Zuo, and I. N. Hajj, "A global driver sizing tool for functional crosstalk noise aviodance," in *Proc. IEEE Int. Symp. Quality Electron. Design*, 2001, pp. 158–163.

[14] T. Stohr, H. Alt, A. Hetzel, and J. Koehl, "Analysis, reduction and avoidance of crosstalk in VLSI design," in *Proc. Int. Symp. Physical Design*, 1998, pp. 211–218.

[15] J. S. Yim and C. M. Kyung, "Reducing crosscoupling among interconnect wires in deep submicron datapath design," in *Proc. Design Automation Conf.*, 1999, pp. 485–490.

[16] H. Zhou and D. F. Wong, "Global routing with crosstalk constraints," in *Proc. Design Automation Conf.*, 1998, pp. 374–377.

[17] S. S. Sapatnekar, "On the chicken-and-egg problem of determining the effect of crosstalk on delay in integrated circuits," in *Proc. IEEE 8th Topical Meeting Elect. Perform. Electron. Packag.*, 1999, pp. 245–248.

[18] G. Hactel, R. Jacoby, and P. Moceyunas, "Performance enhancements in BOLD using implications," in *Proc. Int. Conf. Computer-Aided Design*, 1988, pp. 94–97.

[19] W. Kunz and P. R. Menon, "Multi-level logic optimization by implication analysis," in *Proc. Int. Conf. Computer-Aided Design*, 1994, pp. 6–13.

[20] A. Rubio, N. Itazaki, X. Xu, and K. Kinoshita, "An approach to the analysis and detection of crosstalk faults in digital VLSI circuits," *IEEE Trans. Computer-Aided Design*, vol. 13, pp. 387–395, Mar. 1994.

[21] K. L. Sheppard, "Design methodologies for noise in digital integrated circuits," in *Proc. Design Automation Conf.*, June 1998, pp. 94–99.

[22] R. Arunachalam, K. Rajagopal, and L. T. Pileggi, "Taco: Timing analysis with coupling," in *Proc. Int. Conf. Computer-Aided Design*, Nov. 1998, pp. 1–8.

[23] P. Chen and K. Keutzer, "Toward true crosstalk noise analysis," in *Proc. Int. Conf. Computer-Aided Design*, Nov. 1999, pp. 132–137.

[24] Y. Sasaki and G. De Michelli, "Crosstalk delay analysis using relative window method," in *Proc. Int. ASIC/SoC Conf.*, 1999, pp. 9–13.

[25] P. D. Gross, R. Arunachalam, K. Rajagopal, and L. T. Pileggi, "Determination of worst-case aggressor alignment for delay calculation," in *Proc. Int. Conf. Computer-Aided Design*, Nov. 1998, pp. 1–8.

[26] D. A. Kirkpatrick and A. L. Sangiovanni-Vincentelli, "Digital sensitivity: Predicting signal interaction using functional analysis," in *Proc. Int. Conf. Computer-Aided Design*, 1996, pp. 536–541.

[27] S. Bhardwaj, S. B. K. Vrudhula, and D. Blaauw, "Estimation of signal arrival times in the presence of delay noise," in *Proc. Int. Conf. Computer-Aided Design*, Nov. 2002, pp. 418–422.

[28] W. Feller, *An Introduction to Probability Theory and its Applications*, 3rd ed. New York: Wiley, 1968, vol. 1.

[29] J. Proakis, *Introduction to Digital Communications*. New York: McGraw–Hill, 1983.

[30] H.-Y. Xie and S. B. K. Vrudhula, “A method for estimating signal activities in logic circuits by local propagation of transition probabilities,” *Integrated Computer-Aided Eng.*, vol. 5, no. 2, pp. 141–152, 1998.

**Sarma Vrudhula** (M’85) received the B.Math degree (honors) from the University of Waterloo, Ontario, ON, Canada, in 1976 and the M.S. and Ph.D. degrees in electrical engineering from the University of Southern California, Los Angeles, in 1980 and 1985, respectively.

He is a Professor in the Electrical and Computer Engineering Department at the University of Arizona, Tucson, and the Director of the National Science Foundation Center for Low-Power Electronics. From 2000 to 2001, he was a Visiting Researcher with the Advanced Design Tools Group, Motorola, Austin, TX. His teaching experience includes both undergraduate and graduate courses in digital systems design and testing, very large scale integration (VLSI) design, computer-aided design (CAD) algorithms for VLSI, advanced synthesis and verification methods, computer architecture, and discrete mathematics. His research interests are in design automation and VLSI CAD. These include optimization problems that arise in chip layout, digital systems testing, logic synthesis and verification, energy-efficient system design, and field programmable digital and analog arrays.

Prof. Vrudhula has served on the technical program committees of many national and international conferences in VLSI CAD, on government review panels and as an Associate Editor for the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS.

**David T. Blaauw** (M’01) received the B.S. degree in physics and computer science from Duke University, Durham, NC, in 1986 and the M.S. and Ph.D. degrees in computer science from the University of Illinois, Urbana–Champaign, in 1988 and 1991, respectively.

He was with the Engineering Accelerator Technology Division, IBM Corporation, Endicott, NY, as a Development Staff Member, until August 1993. From 1993 to August 2001, he was with Motorola, Inc., Austin, TX, where he was the Manager of the High Performance Design Technology Group. Since August 2001, he has been a member of the faculty at the University of Michigan, Ann Arbor, as an Associate Professor. His work has focused on VLSI design and CAD with particular emphasis on circuit analysis and optimization problems for high-performance and low-power designs.

Dr. Blaauw was the Technical Program Chair and General Chair for the International Symposium on Low Power Electronics and Design, in 1999 and 2000, respectively, and was the Technical Program Co-Chair and member of the Executive Committee of the ACM/IEEE Design Automation Conference in 2000 and 2001.

**Supamas Sirichotiyakul** received the B.Eng. degree in computer engineering from Chulalongkorn University, Bangkok, Thailand, and the M.S. degree in computer engineering from the University of Louisiana, Lafayette.

From 1995 to 2001, she worked in computer-aided design research and development in the Advanced Tools Group, Motorola, Inc., Austin, TX. She is currently a Member of the Technical Staff in computer-aided design with Sun Microsystems, Inc., Burlington, MA.