

Short Papers

Early Probabilistic Noise Estimation for Capacitively Coupled Interconnects

Murat R. Becer, David Blaauw, Rajendran Panda, and Ibrahim N. Hajj

Abstract—One of the critical challenges in today's high-performance IC design is to take noise into account as early as possible in the design cycle. Current noise analysis tools are effective at analyzing and identifying noise in the postroute design stage when detailed parasitic information is available. However, noise problems identified at this stage of design cycle are very difficult to correct due to the limited flexibility in design and may cause additional iterations of routing and placement which adds costly delays in its time to market. In this paper, we introduce a probabilistic preroute noise analysis approach to identify postroute noise failures before the actual detailed route is completed. We introduce new methods to estimate the RC characteristics of victim and aggressor lines, their coupling capacitances, and the aggressor transition times before routing is performed. The approach is based on congestion information obtained from a global router. Since the exact location and relative position of wires in the design are not yet available, we propose a novel probabilistic method for capacitance extraction. We present results on two high-performance microprocessors in $0.18\ \mu\text{m}$ technology that demonstrate the effectiveness of the proposed approach.

Index Terms—Congestion, crosstalk noise, global routing, probabilistic extraction.

I. INTRODUCTION

Coupling capacitance between neighboring nets is a dominant component in deep submitter designs as taller and narrower lines are being laid out closer to each other [1], [2]. This trend is causing the ratio of cross-coupling capacitance to total capacitance of a wire to increase [2]–[5]. In addition, more aggressive and less noise-immune circuit structures, such as dynamic logic, are now commonly employed due to performance requirements. As a result, a significant crosstalk noise problem has emerged in today's high-performance designs. Crosstalk noise not only leads to modified delays [2], [6], [7] but also to potential logic malfunctions [5], [8]–[11]. In this paper, we focus on the latter, although the presented techniques can also be applied to the former problem as well.

In noise analysis, a victim net is a net on which noise is injected by one or more neighboring nets through coupling capacitances. The nets that inject noise onto a victim net are considered its aggressor nets. In the later stages of the design cycle (i.e. postrouting), detailed information on the topology and relative position of nets is available, making it possible to perform accurate parasitic extraction and noise analysis. Such noise analysis tools typically use linear models of the aggressor and victim driver gates [6], [8], [12] and obtain the noise pulse at the victim receiver input either by solving closed form equations from simplified interconnect models [13]–[19], using simplified circuit analysis

techniques [4], [20], or by solving the resulting linear circuit using reduced-order models [21], [22]. These tools also utilize timing windows calculated during timing analysis [23]–[26] and logic constraints in the circuit [8], [11], [27] when considering which aggressor nets can switch simultaneously to reduce pessimism. Noise from different aggressors of a victim net are combined using linear superposition. The noise tolerance of the receiver gates is usually precharacterized by noise rejection curves of the receiver gate [8], and a noise failure is reported if the noise falls into the failing region of this curve.

If no earlier precautions have been taken, the number of failing nets can be very large, reaching several thousand in current high-performance designs. However, the flexibility to change the design and fix hundreds or even thousands of noise problems in postroute stage is greatly reduced. Noise correction techniques, such as driver sizing [28]–[30], wire spacing, and buffer insertion [31], [32] are difficult to apply and will typically require that the entire design be rerouted. This, however, would drastically alter the location of the nets in the design and can give rise to new noise failures on nets that were previously stable. Solving the crosstalk problem postroute, therefore, can lead to convergence problems and lengthens the design cycle considerably. Although methods have been proposed to solve noise problems during routing [33], [34], these methods typically utilize a limited set of noise correction methods (such as wire spacing and wire sizing). For performance reasons, they also use approximate noise models that will not identify noise failures on all net topologies.

To remedy this situation, methods that allow designers to identify problematic nets in an earlier design stage are required when they can be fixed easily through a host of methods, such as driver sizing, buffer insertion, routing layer assignment, wire sizing and spacing, and receiver gate sizing. However, while much flexibility exists to fix noise at the preroute design stage, only little information is available on which nets are likely to fail. Exact wire length, wire topology, and relative positioning of wires are not available. Therefore, the distributed interconnect characteristics of a net, its coupling capacitance to neighboring nets and the driver strength of its neighbors, which are necessary to perform noise analysis, must be estimated accurately before actual routing is performed.

In this paper, we investigate two possible methods that can be used to estimate interconnect and driver parameters prior to routing, leading to an accurate preroute noise estimation. First, we investigate the use of a simple Steiner tree-based estimated router [35]. This router generates an estimate of the wire length and topology from which an RC representation of the net is constructed. However, no reliable information regarding the location of the neighboring nets is available making it difficult to correctly estimate the cross-coupling capacitance of the net, which is critical for noise analysis. We then propose the use of a global router which provides a more accurate estimate of the routing length and topology of the net and also provides global congestion information which is used to estimate the proximity of neighboring nets. We first look at various correlations between the route obtained from estimated global router and the actual detailed route to verify the consistency between these routes. We then propose a probabilistic method for constructing an approximate coupled interconnect model from the global routing information. The proposed method constructs a coupled interconnect model for a net, segment by segment, using a novel probabilistic extraction approach based on the assignment of wires to global routing cells (g-cells). We also present a method to estimate the aggressor driver strengths, using global routing information. Finally, we

Manuscript received July 29, 2002; revised October 17, 2002. This paper was recommended by Associate Editor S. Sapatnekar.

M. R. Becer and R. Panda are with the Advanced Tools Group, Motorola Inc., Austin, TX 78729 USA (e-mail: MuratBecer@motorola.com).

D. Blaauw is with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109–2122 USA.

I. N. Hajj is with the Coordinated Science Laboratory, University of Illinois, Urbana-Champaign, IL 61801 USA.

Digital Object Identifier 10.1109/TCAD.2002.807892

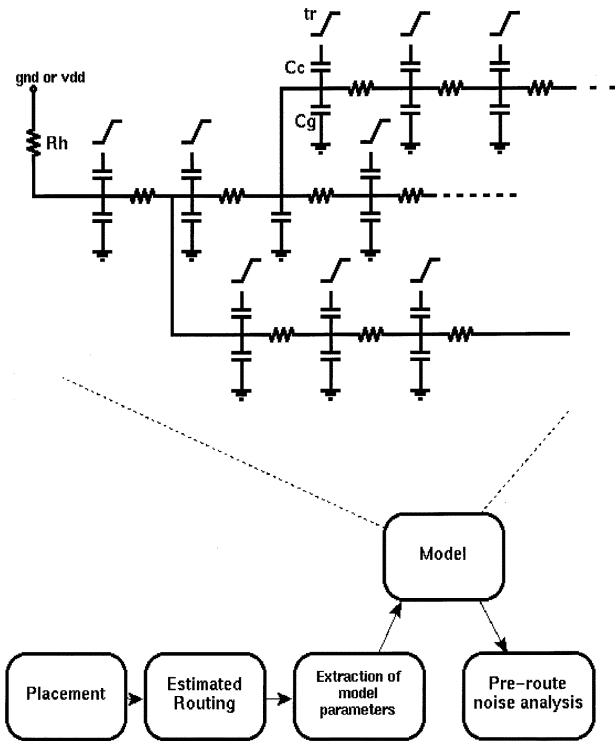


Fig. 1. Noise estimation model.

present results of the two methods on two large industrial processor designs and compare their ability to accurately predict the set of nets that fail postrouting.

The paper is organized as follows. Section II describes the noise estimation methodology and the model we use for preroute noise estimation along with the simple Steiner tree router-based method. Section III presents the properties of global routers and the resulting congestion map. In this section, we also present the congestion-based noise estimation method and our novel probabilistic extraction method. We present results on two high-performance microprocessors in Section IV. Section V contains closing remarks.

II. NOISE ESTIMATION METHODOLOGY

In order to correctly estimate the noise on a net, we construct a model of the net and its aggressor net as shown in Fig. 1.

The victim and aggressor nets are modeled as general RC trees with coupling capacitances between the aggressor and victim nets. The victim driver is represented by an effective holding resistance (R_h). The aggressor driver is represented with an equivalent rise time t_r . We, therefore, need to obtain the following information to estimate the noise on a net: the driver strength of the victim line and aggressor lines, the resistance and grounded capacitance network of the victim and aggressor lines, and the coupling capacitances between them. In order to obtain this information before actual routing has been completed, we use one of several estimated routers as a starting point. Based on the estimated routing information, we compute approximate values for the model parameters and construct the coupled interconnect model shown in Fig. 1. After the model of the interconnect has been constructed, the noise pulse at victim receiver input(s) is calculated using PRIMA [21]. The noise peak and width are then compared against the noise rejection curve of the receiver gate and, accordingly, the net is flagged as failing or not.

The simplest form of estimated routing is a Steiner tree router [35]. This router takes into account one net at a time and does not consider the congestion of the design. Hence, multiple nets can be assigned to

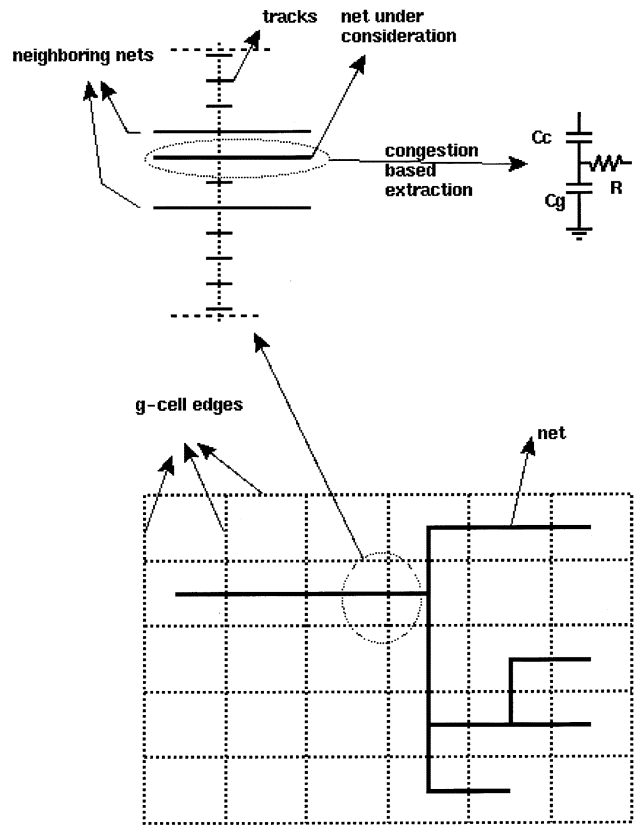


Fig. 2. Congestion map section.

a single track and there is no reliable information regarding the proximity and identity of neighboring nets. Only an estimate of the length and topology of the victim net can be obtained. Based on this estimated victim net topology, an RC tree representation of the net can be constructed with grounded capacitances using calibrated per unit length capacitance and sheet resistance values. The grounded capacitances in this RC netlist represent an estimate of the total net capacitance $C_{total} = C_c + C_g$. The missing parameters required to do preroute noise estimation are therefore coupling capacitances and aggressor transition time information. In order to estimate the coupling capacitance, a portion of each grounded capacitor in the RC netlist is split off and connected as coupling capacitance using a ratio α as follows:

$$C_c = \alpha \times C_{total} \quad (1)$$

$$C_g = (1 - \alpha) \times C_{total}. \quad (2)$$

For the aggressor transition time t_r , a conservative value based on the speed of the design under consideration can be used. Due to the crude nature of these estimations, significant discrepancy can exist between the estimated noise analysis and detailed noise analysis after routing, resulting in either *false failures* or *missed failures*. False failures are nets that are erroneously identified as failing in the estimated noise analysis and require unnecessary allocation of resources to fix them. Missed failures are nets that are erroneously identified as not failing in the estimated noise analysis and, therefore, need to be fixed postrouting, possibly requiring a rerouting and design iterations.

III. CONGESTION-BASED PARAMETER EXTRACTION AND NOISE ESTIMATION

A. Estimated Global Routing

We propose the use of a global router which takes congestion into account as it routes each net. A global router:

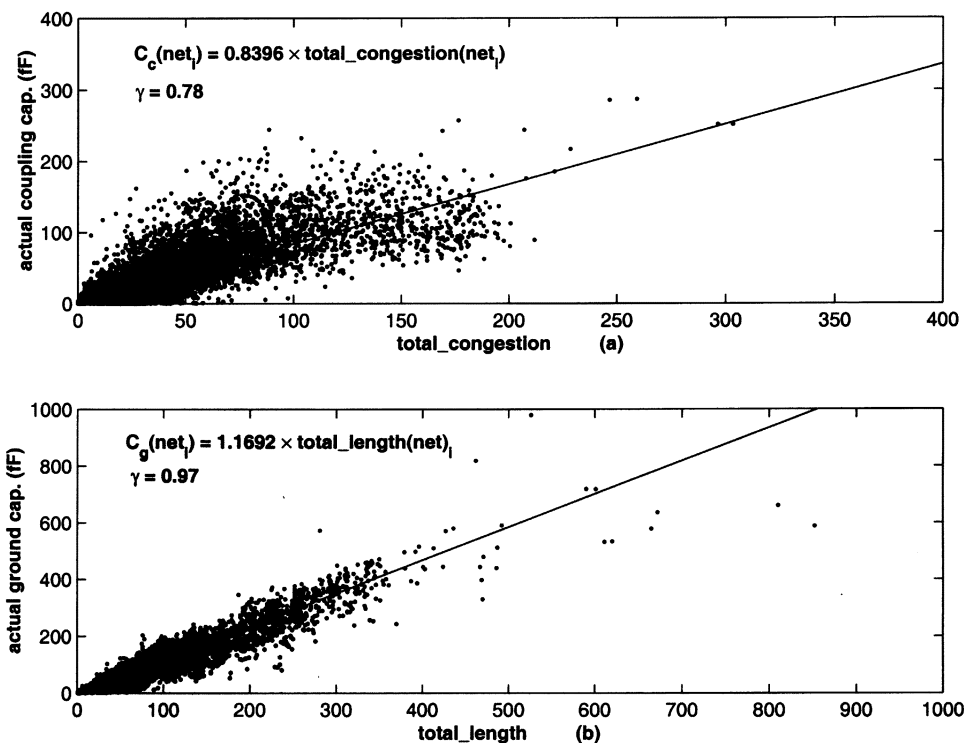


Fig. 3. Correlations between estimated global router and detailed router.

- divides the design into cells;
- determines the number of available tracks for each cell;
- connects the pins of a net utilizing the available tracks of cells while taking congestion into account.

A simplified view of part of the congestion map is shown in Fig. 2. Congestion information is available on a g-cell by g-cell basis. For each g-cell edge, n_i is the total number of tracks in the edge and k_i is the number of tracks used by the global router. The set of nets net_i assigned to each g-cell edge is also available. Note that, although we know n_i , k_i , and the set of nets that are using those k_i tracks, there is no information on which particular track within edge s_i a given net net_i is using (i.e. the global router does not order the nets and, thus, the exact neighbors of a net are still unknown).

Global routers have the advantage that they are very fast while also providing an accurate estimation of the wirelength, topology, and congestion information of a net. They are, therefore, well-suited for early noise estimation before final routing is completed. In this paper, we use an internally developed global router called “Daphne,” although similar results are expected with other global routers. In our noise estimation approach, we use the congestion information from a global router to extract interconnect parameters such as resistance and ground capacitance as well as coupling capacitance and aggressor information for each net. We propose a method that uses a probabilistic approach to extract a coupled interconnect model for each g-cell that a net traverses and then combines these models to form the complete coupled interconnect model for the net. The proposed approach is discussed in the following subsections.

B. Probabilistic Extraction

Since an estimated global router takes congestion into account, it does not overuse the tracks and, in the resulting route, the length and congestion of nets are typically consistent with those after detailed routing. To verify this consistency, we look at correlations between the following values from estimated global routing and detailed routing for

58 000 nets from a high-performance microprocessor in 0.18 μm technology:

- estimated total congestion of a net versus actual extracted coupling capacitance of a net;
- estimated total length of a net versus actual extracted ground capacitance of a net.

Total length of a net is defined as the number of g-cell edges it traverses in the congestion map. Total congestion of a net is defined as follows:

$$\sum_{i=1}^{\text{edges}(\text{net})} \frac{k_i}{n_i} \quad (3)$$

where k_i is the number of nets using g-cell edge i and n_i is the number of available tracks in g-cell edge i .

Fig. 3(a) shows the correlation between total congestion and coupling capacitance. Each dot on the scatter plot corresponds to a net. The line that goes through the scatter plot is a least squares-based linear fit to this data. The correlation coefficient between total congestion obtained from estimated global routing and coupling capacitance extracted from detailed routing is 0.78. Fig. 3(b) shows the correlation between total length and ground capacitance. The correlation coefficient between total length obtained from estimated global routing and ground capacitance extracted from detailed routing is 0.97. As can be seen, there is a strong correlation between these parameters, indicating a consistency in the behaviors of the estimated global router and the detailed router that we use. The global routing information is, therefore, a good source from which to extract the parameters required in Fig. 1 for preroute noise analysis.

We perform a probabilistic estimation of the coupling and grounded capacitances using the congestion information for each g-cell edge that a net traverses. We first characterize the per unit length coupling and ground capacitance values for a particular interconnect technology for a number of density configurations using a field simulator. For example, a net segment is defined to be in a dense region (high congestion) in the congestion map if both of its neighboring tracks are occupied by

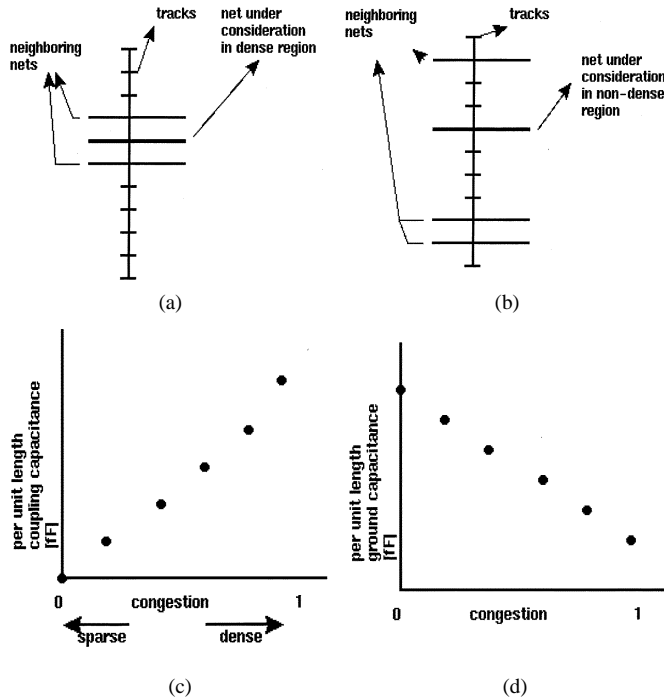


Fig. 4. Dense and nondense configurations on a congestion map edge.

other nets [Fig. 4(a)]. This density decreases as the nearest neighboring nets occupy farther away tracks (lower congestion) [Fig. 4(b)]. Fig. 4(c) shows how the per unit length coupling capacitance and ground capacitance vary with different density configurations. On the x axis, congestion is normalized, 0 represents the most sparse configuration, namely, a net with no close neighbors, and 1 represents the densest possible configuration, namely, a net with both its neighboring tracks occupied by other nets.

In our probabilistic capacitance estimation technique, we estimate the per unit length coupling and grounded capacitance values for a net segment by enumerating the possible density configurations of the net, in each g-cell edge that it traverses. The enumeration is done with respect to all the other nets sharing the same g-cell edge and also with respect to the nets in the neighboring g-cell edges for those density configurations in which the net under consideration is at or close to the g-cell boundaries. The estimated per unit length coupling and grounded capacitances for the net segment under consideration is then found by taking a weighted average of per unit length capacitance values shown in Fig. 4(c), weighted by the probability of each density configuration which is computed through their corresponding number of enumerations. This technique allows one to estimate both the coupling and grounded capacitances for the net segment that traverses a specific g-cell edge in the congestion map. As a result, this approach provides a different α value for each net segment in the congestion map.

We now quantify the approach as follows. The total number of possible configurations for a g-cell edge is

$$\text{total configurations} = \binom{n}{k} k! \quad (4)$$

where n is the number of tracks in the edge and k is the number of utilized tracks. It is infeasible to enumerate all the possible density configurations and find all corresponding per unit length capacitances. For this reason we make the following simplifications.

- Capacitance values of a net are affected only by the location of the nearest neighboring nets.

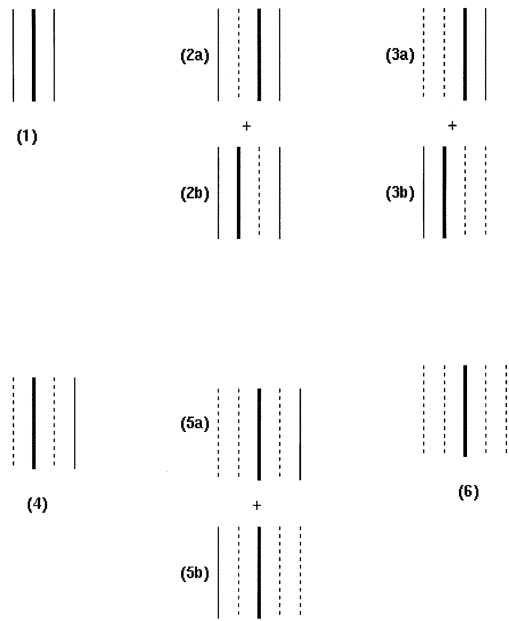


Fig. 5. Configurations used.

- The effect of a neighboring net that is more than two tracks from the net under consideration is considered insignificant.

The first assumption is valid since the closest neighbors act as shields to all other nets. The second assumption is, in general, also valid since the field lines between two nets vanish as their separation increases. From our experiments, coupling between nets that are separated by two or more empty tracks is small. As a result, we need to consider six unique configurations, as shown in Fig. 5.

The bold net is the net under consideration. Dotted lines represent empty tracks. Configuration (1) is the dense configuration where both neighboring tracks are used. In configuration (2), there is an empty track on one side and a dense track on the other side. Configuration (3) represents the case where one side is dense and there are at least two empty tracks on the other side. Configuration (4) has one empty track on both sides, whereas configuration (5) has one empty track on one side and at least two empty tracks on the other side. Finally, configuration (6) has at least two empty tracks on each side. Configurations (1), (2), and (4) represent track assignments with an exactly specified spacing to the nearest neighbors. All net permutations that correspond to these configurations, therefore, have the same per unit length capacitances (based on the first simplification stated above). Configurations (3), (5), and (6) represent track assignments with a range of densities, since the spacing of the nearest neighbor on one or both sides can vary. Since this neighboring net has a spacing of at least two or more tracks, the capacitance contributed by this neighbor net is small. We, therefore, use the average of the minimum per unit length capacitance (one side dense, the other side has no neighbors at all) and maximum per unit length capacitance (one side dense, the other side with two empty tracks and then a neighbor) for these configurations. Note that the per unit length capacitances for the configurations shown in Fig. 5 have been calculated with the assumption that the orthogonal layers have a fixed congestion. A possible extension to our approach is to increase the number of configurations such that the varying congestion in orthogonal metal layers is also taken into account.

After determining the per unit length coupling and grounded capacitance for each configuration in Fig. 5, we compute probability of each configuration. The probability of configuration i is defined as

$$\text{prob}_{(i)} = \frac{\text{conf}_{(i)}}{\text{total configurations}} \quad (5)$$

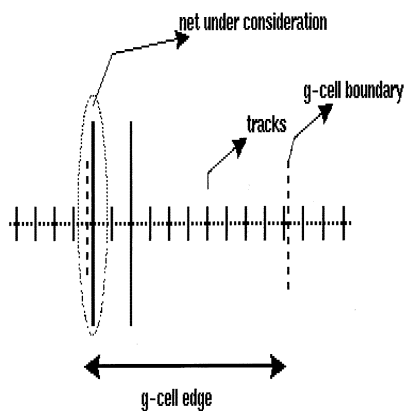


Fig. 6. Net on the boundary track.

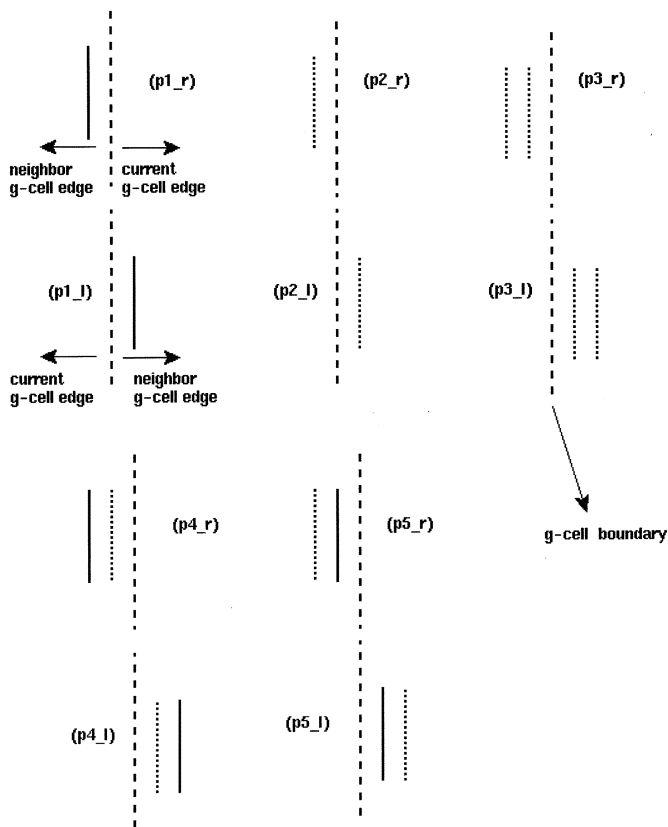
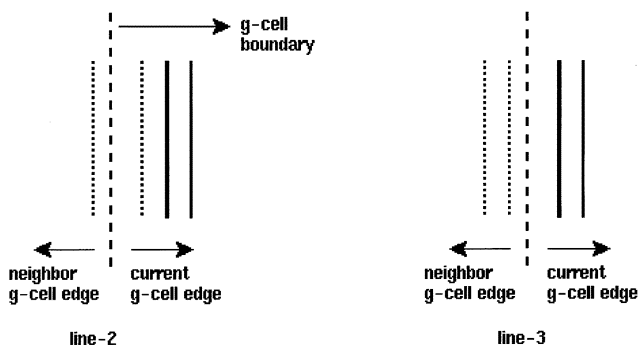


Fig. 7. Neighbor edge congestion.

where $conf_{(i)}$ is the number of enumerations that correspond to configuration i . The number of enumerations for each configuration depends on the number of tracks available n_i and the number of nets in this g-cell edge k_i . One should pay attention to the special cases when the net under consideration is close to the boundaries of the g-cell edge. For example, Fig. 6 shows a configuration where the net segment under consideration (bold net) is on the boundary track of the g-cell edge with one empty and one filled track to its right. There will be many enumerations (permutations) in which the configuration in Fig. 6 will be valid. The decision of how to distribute these enumerations among the six defined density configurations in Fig. 5, depends on the left neighbor track of the net segment under consideration. Thus, the congestion of the neighboring g-cell edges should also be taken into account to be able to accurately make decisions on such cases. For this purpose, we find the probabilities of the configurations shown in Fig. 7.

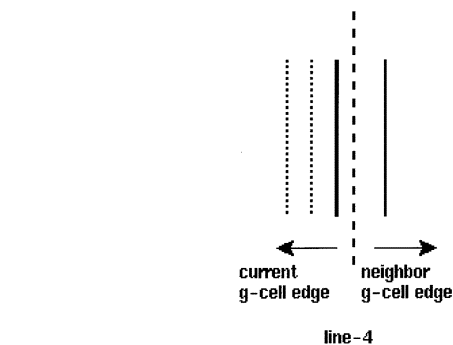


Fig. 8. Special cases for configuration (3a).

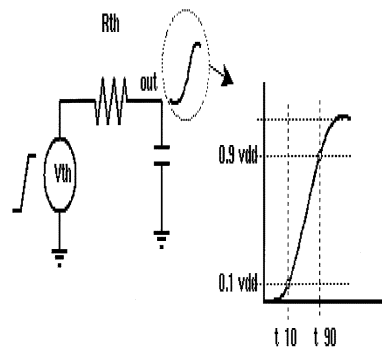
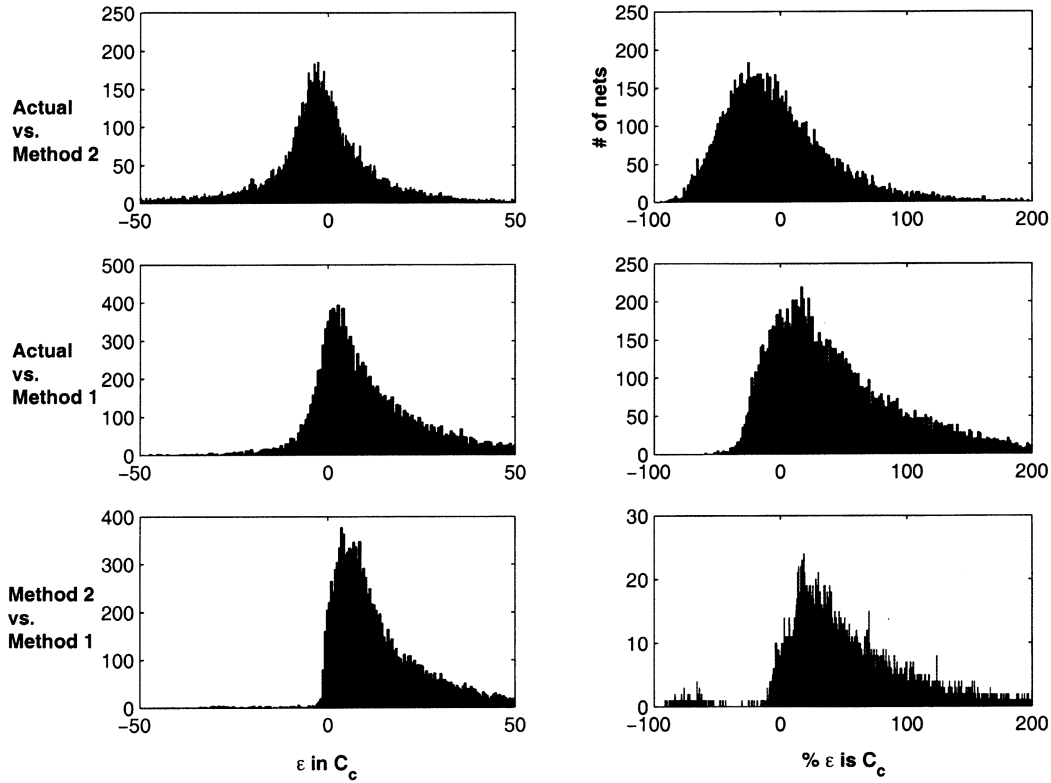


Fig. 9. Model to find t_r of aggressor line.

In Fig. 7, $p1_r$ is the probability that the right most track of a g-cell edge is filled. On the other hand, $p2_r$ is the probability that the right most track of a g-cell edge is empty. If the number of tracks is n and the number of nets using this g-cell edge is k , then $p1_r$ is calculated as follows:

$$p1_r = \frac{k \binom{n-1}{k-1} (k-1)!}{\binom{n}{k} k!} = \frac{k}{n} \tag{6}$$

The denominator of (6) is the total number of enumerations in this g-cell edge. The numerator is the number of enumerations where the rightmost track of this g-cell edge is filled. We can choose the net on the rightmost track among k nets. The remaining $k - 1$ nets can be anywhere among the remaining $n - 1$ tracks in any order, thus the last two terms in the numerator. The other probabilities in Fig. 7 are also calculated similarly. Also note that that same argument applies for horizontal g-cell edges where the nets will have ordering in the vertical axis.

Fig. 10. Error in C_c .

When we look at Fig. 6, we can see that these enumerations may fall into the category of configuration (2b) in Fig. 5 with a probability of $p1_r$ (Fig. 7), configuration (4) with a probability of $p4_r$, and configuration (5a) with a probability of $p3_r$. As a result, the number of enumerations in Fig. 6 are distributed into three configurations based on the neighboring g-cell edge congestion.

As examples, we demonstrate the calculation of the number of enumerations for two configurations, (1) and (3a). For configuration (1), the number of enumerations is

$$\begin{aligned}
 \text{conf}_{(1)} &= (n-2) \times \binom{k-1}{2} \times 2 \\
 &\times \binom{n-3}{k-3} \times (k-3)! \\
 &+ (k-1) \times \binom{n-2}{k-2} \times (k-2)! \\
 &\times (p1_r + p1_l). \tag{7}
 \end{aligned}$$

We can explain (7) as follows. For the victim net to be in a dense region within the edge, it can be anywhere except for the boundary tracks of the g-cell edge under consideration. There are $(n-2)$ such tracks. There needs to be two nets in its neighboring tracks. These two nets can be chosen among the remaining $k-1$ nets and can be in any order, resulting in the second and third factor in (7). The rest of the nets should be placed on the remaining tracks. There are $k-3$ nets and $n-3$ tracks left, and the nets can be in any order, resulting in the two factors in the second line in (7). The last two lines are for the special cases where the victim net is on the boundary tracks. The third line gives the number of such enumerations. We should have one net among $k-1$ nets as one neighbor within the edge and the remaining $k-2$ nets should be placed in the remaining $n-2$ tracks. Note that the number of such enumerations for either side of the edge is equal.

TABLE I
PARAMETER ESTIMATION ERRORS

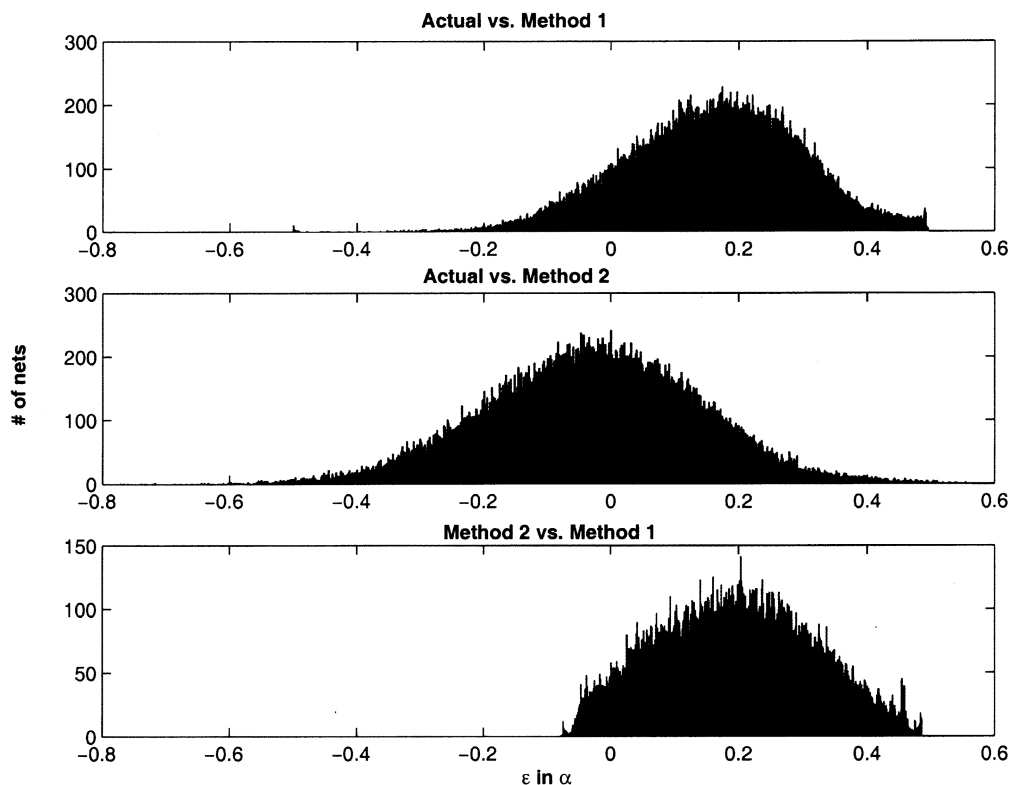
		Chip-1	Chip-2
Actual	C_c (fF/%)	19.8/55	21.4/86
vs.	C_g (fF/%)	18.6/32.7	18.9/31.1
Steiner	α (%)	0.18/110	0.24/227
Actual	C_c (fF/%)	9.8/24	12/33
vs.	C_g (fF/%)	10.4/18.2	12.5/21
Probabilistic	α (%)	0.1/42	0.11/48
Probabilistic	C_c (fF/%)	19.2/51.4	22.5/68
vs.	C_g (fF/%)	12.4/21.2	12.8/22
Steiner	α (%)	0.2/114	0.23/170

The probability that these enumerations can be added to configuration (1) is the probability that there will be a filled track on the immediate neighboring track which is a part of the neighboring g-cell edge. This is represented by the last line of (7).

If we look at configuration (3a), the number of enumerations where one side is dense and the other side has at least two empty tracks is

$$\begin{aligned}
 \text{conf}_{(3a)} &= (n-3) \times (k-1) \times \binom{n-4}{k-2} (k-2)! \\
 &+ (k-1) \times \binom{n-3}{k-2} (k-2)! \times p2_r \\
 &+ (k-1) \times \binom{n-2}{k-2} (k-2)! \times p3_r \\
 &+ \binom{n-3}{k-1} (k-1)! \times p1_l. \tag{8}
 \end{aligned}$$

The first line in (8) represents the number of enumerations that would fall into category (3a) within the g-cell edge. The last three lines in (8) are due to special cases where the victim net can be on the boundary

Fig. 11. Error in α .

or one track away from the boundary of the cell. For clarity, we show these cases in Fig. 8. The number of enumerations for all other cases in Fig. 5 can be calculated similarly.

For each edge that a net traverses, we compute the weighted contribution of the six possible configurations as explained above. Each configuration has a precomputed per unit length coupling capacitance $cc_{(i)}$, and a per unit length grounded capacitance $cg_{(i)}$ computed with a field solver (Fig. 4). The contribution of configuration (i) to the coupling and grounded capacitances for the net segment is

$$cc_{\text{total}(i)} = cc_{(i)} \times \frac{\text{conf}_{(i)}}{\text{total configurations}} \quad (9)$$

$$cg_{\text{total}(i)} = cg_{(i)} \times \frac{\text{conf}_{(i)}}{\text{total configurations}}. \quad (10)$$

The total coupling and ground capacitances of a net segment is finally calculated by summing the weighted contributions for all configurations and scaling the per unit length values with the length of the net segment. Once we have obtained the probabilistic grounded capacitance and coupling capacitance values for each edge that a net traverses, a coupled RC circuit representation is constructed, as illustrated in Fig. 1.

C. Aggressor Strength Estimation

Methods explained so far let us estimate the victim net parameters and coupling capacitances for each net (Fig. 1). To be able to perform estimated noise analysis, we should also estimate aggressor net characteristics for each net (represented as t_r in Fig. 1). A simple and conservative solution would be to use the strongest transitional net in the circuit as the default aggressor. Such an approach is likely to cause many false failures.

We can use the congestion information to estimate an average aggressor for each net. All nets that share a g-cell edge are possible neigh-

TABLE II
FAILING NETS IN PRE- AND POSTROUTE NOISE ANALYSIS

Chip	Method	Missed	Common	False
chip-1	Steiner	148	694	531
chip-1	Probabilistic	161	681	205
chip-2	Steiner	134	487	3509
chip-2	Probabilistic	90	531	1409

bors. The likelihood of a net to be an aggressor to another net increases as the number of shared edges increases. To estimate an average aggressor for a net, we first find the ten possible neighbors with the highest number of shared edges. For each of these nets, we apply the following procedure to find their signal transition time $t_{r_{\text{out}}}$

- Find the total capacitance C_t of the net, using methods explained in previous sections.
- Obtain the Thevenin model of the driver gate using precharacterized information from the cell library.
- Compute the transition time $t_{r_{\text{out}}}$ using the model shown in Fig. 9

The normalized time domain solution for the voltage at node out is as follows:

$$v_{\text{out}}(t) = \begin{cases} \frac{-RC+t+RCe^{-t/RC}}{t_r} & 0 < t \leq t_r \\ \frac{-RC}{t_r}(e^{t_r/RC} - 1)e^{-t/RC} + 1 & t > t_r \end{cases}. \quad (11)$$

Although it is not possible to solve (11) analytically, the rise time at node out can easily be computed using binary search. After finding the 10% and 90% time points at node out (Fig. 9), $t_{r_{\text{out}}}$ is found using extrapolation as shown in (12). An average aggressor transition time for a victim net is then found as the weighted average (weighted by the number of shared edges) of the $t_{r_{\text{out}}}$ values for all nets considered as its possible neighbors.

$$t_{r_{\text{out}}} = \frac{t_{90} - t_{10}}{0.8} \quad (12)$$

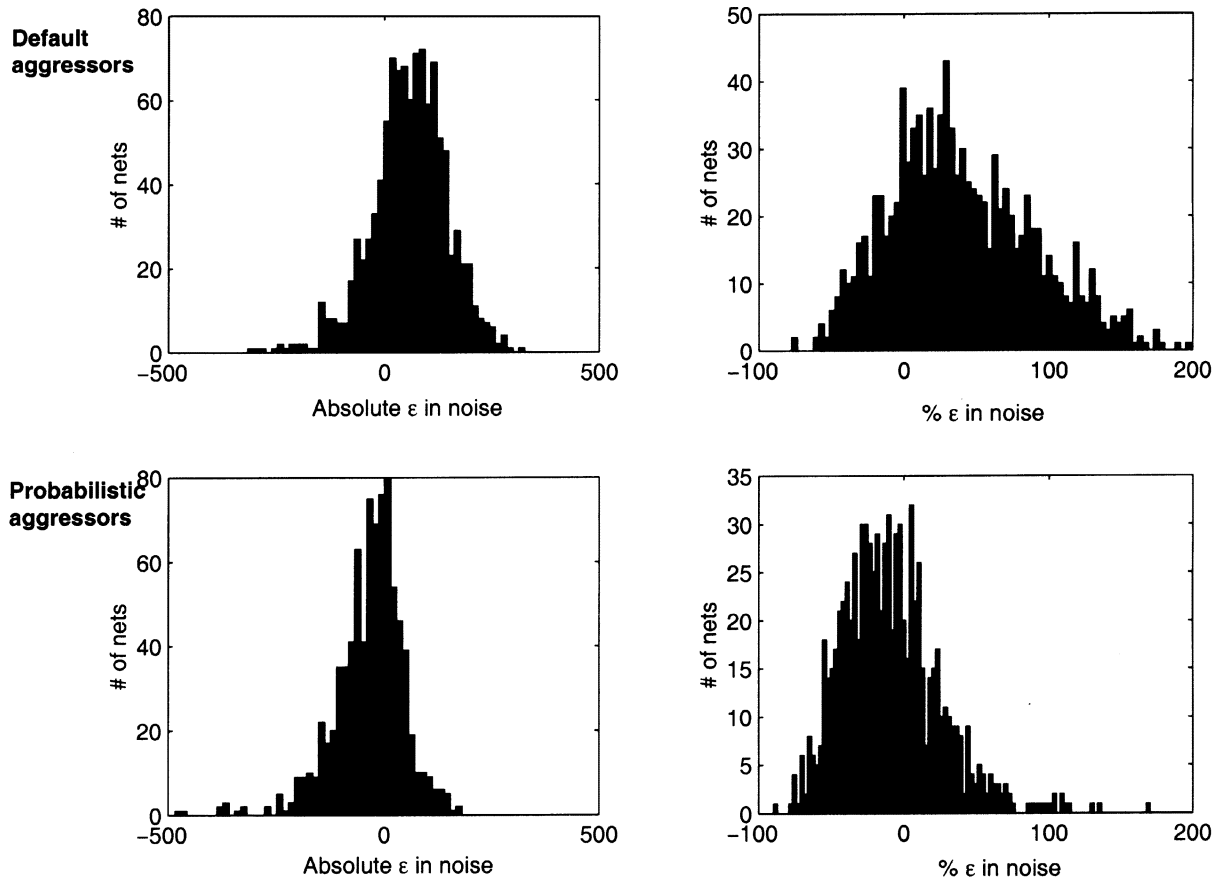


Fig. 12. Default versus Probabilistic aggressors.

IV. RESULTS

In this section, we present results on two high-performance microprocessors in $0.18\ \mu\text{m}$ technology. Chip one has 58 000 nets, whereas chip two has 125 000 nets. We first look at how the methods presented in this paper estimate parameters such as total coupling capacitance in the preroute stage. Fig. 10 shows the errors in the coupling capacitances using the two methods described. Method one refers to the Steiner tree routing based approach with an α ratio of 0.5. Method two refers to the probabilistic extraction method described in Section III-B. The first column of graphs shows the absolute errors in femtofarads, whereas the second column of graphs shows the percentage errors. The first two rows in Fig. 10, present the errors in C_c using the two methods, with respect to postroute extracted C_c for each net. The last row compares the C_c estimations of the two methods, confirming the overestimation obtained by the Steiner method in comparison to the congestion-based probabilistic method. The average absolute and percentage errors, in magnitude, for several estimated parameters are shown in Table I. For each method, average absolute and percentage errors in estimated parameters, i.e., coupling capacitance (C_c), grounded capacitance (C_g) and α (ratio of coupling capacitance to total capacitance) are presented in the following format for the two chips: average absolute error / average percentage error. Note that the absolute errors for capacitances are in femtofarads, whereas the absolute error in α is unitless. These errors are calculated by comparing early parameter estimations with postroute extracted data for the first two rows, whereas the last row shows a comparison between the two methods. As can be seen, using congestion information provides more accurate model parameters than Steiner tree routing.

Fig. 11 shows the absolute errors in α for chip one using the two methods, where α is the ratio of the total coupling capacitance to the

total capacitance of a net. The error is relative to the α obtained for extracted nets. As can be seen, the probabilistic extraction method provides significantly better α ratios. The reason for this is that probabilistic extraction method extracts localized α values for each g-cell edge that a net is traversing in the congestion map.

In Table II, we present the number of failing nets found after estimated noise analysis using the described method in Section II. We compare these results with the number of failures using postroute noise analysis on the same design after detailed routing and full extraction. Detailed routing and extraction has been done using commercial tools. An inhouse tool ClariNet [8] has been used to perform post route noise analysis. The column named "common" shows the number of nets that fail in both pre- and postroute noise analysis. The column named "missed" shows the number of nets not identified by preroute noise analysis that subsequently failed in postroute noise analysis. The column named "false" shows the number of nets that failed in preroute noise analysis but not in postroute noise analysis. Ideally, the number of missed and false failures identified by the preroute noise analysis is zero. False failures unnecessarily increase the required design resources to fix noise problems before routing. Missed failures, on the other hand, are especially detrimental since they require noise fixes after routing which can require rerouting and additional design cycles.

Table II shows that the Steiner tree-based method has dramatically more false failures than the congestion-based method, whereas the difference in number of missed nets is relatively small. Using the probabilistic method reduces the number of false failures by as much as 60%, or 2100 nets.

Finally, we look at the effects of using a default a aggressor versus using our probabilistic estimation of aggressors as we perform estimated noise analysis. In addition to identifying postroute problematic

nets early in the design cycle, we also need to estimate the actual noise amount on these nets as accurately as possible. Since the results of preroute noise estimation is to be used for noise avoidance purposes, estimating the amount of noise to be reduced on each net will help noise avoidance algorithms produce correct results. Using the probabilistic aggressor estimation, the number of "common" nets are reduced by 8% compared to a strong default aggressor. But on the other hand, the average error in estimated noise peaks with respect to actual noise peaks from postroute noise analysis go down from 40% to 9%. Fig. 12 shows the error in the estimated coupling noise peaks on the noisy nets of chip one. The first column shows the absolute errors in mV, where error is defined to be postroute analysis noise—preroute analysis noise for each net. The second column shows the same information in percentages.

V. CONCLUSION

In this paper, we presented preroute noise analysis methods using estimated routing. We showed the close correlation between the detailed router and the estimated global router that we used and proposed methods to extract interconnect parameters and coupling capacitances for each net using congestion information. We proposed a congestion-based probabilistic method for parameter extraction as well as a method for aggressor strength estimation. Results show that a good preroute coupled extraction for each net can be made using our congestion-based method. We also showed that a preroute noise analysis can be performed using this estimated extraction for each net, providing the designer with a high percentage of postroute problematic nets in this early design stage. Methods presented in this paper can be used in identifying future noisy nets at a very early design stage with minimal overhead, estimating noise amounts on these nets, and, ultimately, as input to noise avoidance algorithms which can guide detailed routers to avoid the identified problems.

REFERENCES

- [1] Semiconductor Industry Association, The International Technology Roadmap for Semiconductors, 1999.
- [2] D. Sylvester and K. Keutzer, "Getting to the bottom of deep submicron," in *Proc. IEEE Int. Conf. Computer-Aided Design*, 1998, pp. 203–211.
- [3] Y. Cao, T. Sato, X. Huang, C. Hu, and D. Sylvester, "New approaches to noise aware static timing analysis," in *Proc. IEEE TAU Int. Workshop Timing Issues Specification Synthesis Digital Syst.*, Dec. 2000, pp. 8–13.
- [4] A. Devgan, "Efficient coupled noise estimation for on-chip interconnects," in *Proc. IEEE Int. Conf. Computer-Aided Design*, 1997, pp. 147–153.
- [5] K. L. Shepard and V. Narayanan, "Noise in deep submicron digital design," in *Proc. IEEE Int. Conf. Computer-Aided Design*, Nov. 1996, pp. 524–531.
- [6] S. Sirichotiyakul, D. Blaauw, C. Oh, R. Levy, V. Zolotov, and J. Zuo, "Driver modeling and alignment for worst-case delay noise," in *Proc. Design Automation Conf.*, June 2001, pp. 720–725.
- [7] G. Yee, R. Chandra, V. Ganesan, and C. Sechen, "Wire delay in presence of crosstalk," in *Proc. IEEE TAU Meeting Timing Issues Digital Syst.*, 1997.
- [8] S. Alwar, D. Blaauw, A. Dasgupta, A. Grinshpon, R. Levy, C. Oh, B. Orshav, S. Sirichotiyakul, and V. Zolotov, "ClariNet: a noise analysis tool for deep submicron design," in *Proc. 37th Annu. Design Automation Conf.*, June 2000, pp. 233–238.
- [9] J. M. Zurada, Y. S. Joo, and S. V. Bell, "Dynamic noise margins of mos logic gates," in *Proc. IEEE Int. Symp. Circuits Syst.*, 1989, pp. 1153–1156.
- [10] K. L. Shepard, V. Narayanan, P. C. Elmendorf, and G. Zheng, "Global harmony: coupled noise analysis for full-chip rc interconnect networks," in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design*, Nov. 1997, pp. 139–146.
- [11] K. L. Shepard, "Design methodologies for noise in digital integrated circuits," in *Proc. Design Automation Conf.*, 1998, pp. 94–99.
- [12] F. Dartu, N. Menezes, and L. T. Pileggi, "Performance computation for precharacterized CMOS gates with rc loads," *IEEE Trans. Computer-Aided Design*, vol. 15, pp. 544–553, May 1996.
- [13] J. Cong, D. Zhingang, and P. V. Srinivas, "Improved crosstalk modeling for noise constrained interconnect optimization," in *Proc. Asia South Pacific Design Automation Conf.*, 2001, pp. 373–378.
- [14] M. R. Becer and I. N. Hajj, "An analytical model for delay and crosstalk estimation with application to decoupling," in *Proc. IEEE Int. Symp. Quality Electron. Design*, Mar. 2000, pp. 51–57.
- [15] M. R. Becer, D. Blaauw, V. Zolotov, R. Panda, and I. N. Hajj, "Analysis of noise avoidance techniques in dsm interconnects using a complete crosstalk noise model," in *Proc. Design Automation Eur.*, Mar. 2002, pp. 456–463.
- [16] A. B. Kahng, S. Muddu, and D. Vidhani, "Noise and delay uncertainty studies for coupled rc interconnects," in *Proc. 12th Annu. IEEE Int. ASIC/SOC Conf.*, 1999, pp. 3–8.
- [17] T. Xue, E. S. Kuh, and D. Wang, "Post global routing crosstalk risk estimation and reduction," in *Proc. IEEE Int. Conf. Computer-Aided Design*, 1994, pp. 616–619.
- [18] T. Sakurai, "Closed-form expression for interconnect delay, coupling, and crosstalk in VLSIs," *IEEE Trans. Electron Devices*, vol. 40, pp. 118–124, Jan. 1993.
- [19] A. Vittal, L. H. Chen, M. Marek-Sadowska, K. P. Wang, and S. Yang, "Crosstalk in VLSI interconnections," *IEEE Trans. Computer-Aided Design*, vol. 18, pp. 1817–1824, Dec. 1999.
- [20] M. Kuhlmann and S. S. Sapatnekar, "Exact and efficient crosstalk estimation," *IEEE Trans. Computer-Aided Design*, vol. 20, pp. 858–866, July 2001.
- [21] A. Odabasioglu, M. Celik, and L. T. Pileggi, "PRIMA: passive reduced-order interconnect macromodeling algorithm," *IEEE Trans. Computer-Aided Design*, vol. 17, pp. 645–653, Aug. 1998.
- [22] L. T. Pillage and R. A. Rohrer, "Asymptotic waveform evaluation for timing analysis," *IEEE Trans. Computer-Aided Design*, vol. 9, pp. 352–366, Apr. 1990.
- [23] S. Sapatnekar, "Capturing the effect of crosstalk on delay," in *Proc. 13th Int. Conf. VLSI Design*, Jan. 2000, pp. 364–369.
- [24] R. Arunachalam, K. Rajagopal, and L. T. Pileggi, "Taco: timing analysis with coupling," in *Proc. Design Automation Conf.*, June 2000, pp. 266–269.
- [25] P. Chen, D. A. Kirkpatrick, and K. Keutzer, "Switching window computation for static timing analysis in the presence of crosstalk," in *Proc. IEEE Int. Conf. Computer-Aided Design*, 2000, pp. 331–337.
- [26] Y. Sasaki and G. De Micheli, "Crosstalk delay analysis using relative window method," in *Proc. 12th Annu. IEEE Int. ASIC/SOC Conf.*, 1999, pp. 9–13.
- [27] P. Chen and K. Keutzer, "Toward true crosstalk noise analysis," in *Proc. Design Automation Conf.*, June 1999, pp. 132–138.
- [28] I. H. R. Jiang, Y. W. Chang, and J. Y. Jou, "Crosstalk driven interconnect optimization by simultaneous gate and wire sizing," *IEEE Trans. Computer-Aided Design*, vol. 19, pp. 999–1010, Sept. 2000.
- [29] T. Xiao and M. Marek-Sadowska, "Gate sizing to eliminate crosstalk induced timing violation," in *Proc. Int. Conf. Computer Design*, 2001, pp. 186–191.
- [30] —, "Crosstalk reduction by transistor sizing," in *Proc. Asia South Pacific Design Automation Conf.*, 1999, pp. 137–140.
- [31] 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.
- [32] C. P. Chen and N. Menezes, "Noise aware repeater insertion and wire sizing for on-chip interconnect using hierarchical moment matching," in *Proc. Design Automation Conf.*, 1999, pp. 502–506.
- [33] H. Zhou and D. F. Wong, "Global routing with crosstalk constraints," in *Proc. Design Automation Conf.*, 1998, pp. 374–377.
- [34] A. Vittal and M. Marek-Sadowska, "Crosstalk reduction for VLSI," *IEEE Trans. Computer-Aided Design*, vol. 16, pp. 290–298, Mar. 1997.
- [35] D. Blaauw, A. Devgan, and A. Dharchoudhury, "Signal integrity in high performance design," presented at the Int. Conf. Computer-Aided Design, 1999.