Pre-route Noise Estimation in Deep Submicron Integrated Circuits

Murat R. Becer, David Blaauw*, Rajendran Panda and Ibrahim N. Hajj**
Advanced Tools Group, Motorola Inc., Austin, TX
*EECS Dept. University of Michigan, Ann Arbor, **CSL, University of Illinois at Urbana-Champaign

Abstract

One of the critical challenges in today's high performance IC design is to take noise into account as early as possibe in the design cycle. Current noise analysis tools [1, 7] are effective at analyzing and identifying noise in the post-route design stage when detailed parasitic information is available. However, noise problems identified at this stage of design cycle are very difficult to fix due to the limited flexibility in the design and may cause additional iterations of routing and placement, adding costly delays to time-to-market. In this paper, we introduce an estimated, congestion-based pre-route noise analysis approach to identify post-route 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 is not yet available at this point, we propose a novel probabilistic method for capacitance extraction. We present results on two high performance microprocessors in 0.18μ technology that demonstate the effectiveness of the proposed approach.

1. Introduction

Coupling capacitance between neighboring nets is a dominant component in deep submicron designs as taller and narrower lines are being laid out closer to each other [2]. This trend is causing the ratio of cross-coupling capacitance to total capacitance of a wire to increase. 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 is emerging in today's high performance designs. Crosstalk noise not only leads to modified delays [8] but also to potential logic malfunctions [1, 7]. In this paper, we focus on the latter, although the presented techniques can also be applied for 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. post-routing), 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 and obtain the noise pulse at the victim receiver input either by analytical equations from simplified interconnect models [4, 5] or by solving the resulting linear circuit using a reduced order model such as PRIMA [6]. Noise from different aggressors of a victim net are combined using linear superposition. The noise tolerance of receiver gates is usually pre-characterized by noise rejection curves [1], the combined noise is propagated through the receiver gate, 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 thousands in current high performance designs. However, the flexibility to change the design and fix hundereds or even thousands of noise problems in post-routing is greatly reduced. Noise avoidance techniques, such as driver sizing, wire spacing and buffer insertion, are difficult to apply and will typically require that the entire design is re-routed. 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 post-route therefore can lead to convergence problems and lenghens the design cycle considerably. Although methods have been proposed to solve noise problems during the routing itself [10, 9], these methods typically utilize a limited set of noise avoidance 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 pre-route 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 neigbors, which are necesarry to perform noise analysis must be estimated accurately before actual routing is performed.

In this paper, we investigate three possible methods that can be used to estimate interconnect and driver parameters prior to routing, leading to an accurate pre-route noise estimation. First, we investigate the use of a simple Steiner tree based estimated router [3]. This router generates an estimate of the wire length and topology from which an RC representation of the net is constructed. How-

Proceedings of the International Symposium on Quality Electronic Design (ISQED'02)
0-7695-1561-4/02 $17.00 © 2002 IEEE

ever, no 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 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 actual detailed route to verify the consistency between these routes. We then propose two methods for constructing an approximate coupled interconnect model from the global routing information. The first method uses a calibration based technique to estimate the total grounded and coupling capacitance of a net based on its wire length and congestion. The second 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. We also present a method to estimate the aggressor driver strengths, using congestion information. Finally, we present results of all three methods on two industrial processor designs and compare their ability to predict the set of nets that fail post-routing.

The paper is organized as follows. Section 2 describes the noise estimation methodology and the model we use for pre-route noise estimation along with the simple Steiner tree router based method. Section 3 presents the properties of global routers and the resulting congestion map. In this section, we also present the two congestion based noise estimation methods. We present results on two high performance microprocessors in Section 4. Section 5 contains closing remarks.

2. 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 Figure 1. The

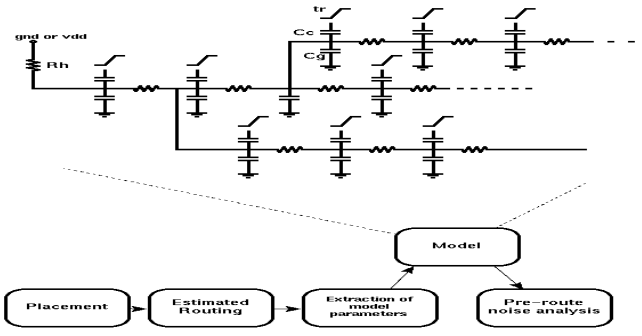

Figure 1. Noise estimation model

victim net 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 as a starting point one of several estimated routers. Based on the estimated routing information, we compute approximate values for the model parameters and construct the coupled interconnect model shown in Figure 1. After the model of the interconnect has been constructed, the noise pulse at victim receiver input(s) is calculated using PRIMA [6]. The noise peak and noise 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 [3]. This router takes one net into account at a time and does not consider the congestion of the design. Hence, multiple nets can be assigned to a single track and there is no reliable information regarding the proximity and identity of neighboring nets, and 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. With this approach, the missing parameters required to do pre-route noise estimation are therefore coupling capacitances and aggressor transition time information. In order to estimate the coupling capacitance, typically 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 post-routing, possibly requiring a re-routing and design iterations.

3. Congestion Based Parameter Extraction and Noise Estimation

3.1. Estimated Global Routing

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

- Divides the design into cells
- Assigns the number of available tracks for each cell
- Connects the instance pins of a net utilizing available tracks of cells while taking congestion into account

A simplified view of part of the congestion map is shown in Figure 2. Congestion information is given on a cell or segment by segment basis. For each segment, n_i is the total number of tracks in the segment and k_i is the number of tracks used by the global router. For each segment, the set of nets net_i assigned to that segment 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 segment 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). In our noise estimation approach, we use

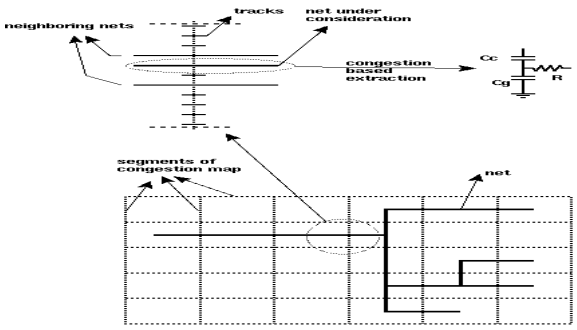

Figure 2. Congestion map section

the congestion map information to extract interconnect parameters such as resistance and ground capacitance as well as coupling capacitance and aggressor information for each net. We propose two methods. The first method uses a calibrated approach based on the total wire length and the average congestion of a net. The second method uses a probabilistic approach to extract a coupled interconnect model for each segment that a net traverses and then combines these models to form the complete coupled interconnect model for the net. The two approaches are discussed in the following two sections.

3.2. Calibration Method

Since an estimated global router takes congestion into account, it does not over use 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 58000 nets from a high performance microprocessor in 0.18μ technology:

- Estimated total congestion of a net vs. actual extracted coupling capacitance of a net
- Estimated total length of a net vs. actual extracted ground capacitance of a net

Total length of a net is defined as the number of segments it goes through in the congestion map. Total congestion of a net is defined as follows:

$$\sum_{i=1}^{segments(net)} \frac{k_i}{n_i} \tag{3}$$

Figure 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. Figure 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 behaviours 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 Figure 1 for pre-route noise analysis.

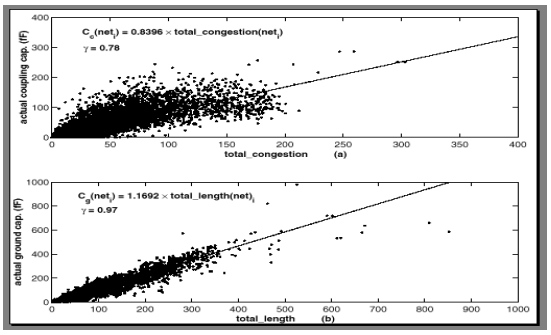

Figure 3. Correlations between estimated global router and detailed router

The fits in Figure 3(a) and 3(b) can be obtained if both estimated global routing data and detailed routing data is available for a design. Since our goal is to estimate the model parameters for pre-route noise estimation, detailed routing data will not be available. In our approach, we therefore first analyze a completed design that is similar to the design under consideration. For instance, for the analysis of a microprocessor, a previous generation processor can be used in an older technology. Based on the detailed and global routing data of the completed design, we obtain the linear fitting coefficients shown in Figure 3. We then scale these linear fit coefficients from the technology of the completed design to the technology of the design under consideration as follows. For a given interconnect technology, we generate the avegare per unit length coupling capacitance $C_{c_{unit}}(t_i)$ and average per unit length ground capacitance $C_{g_{unit}}(t_i)$, using a field simulator. These values are found by averaging the per unit length capacitances for multiple interconnect structures with different wire widhts and spacings. Given the linear coefficient for coupling capacitance $cc_{coef}(t_i)$ and ground capacitance $cg_{coef}(t_i)$ for a design in interconnect technology t_i, we compute the total coupling and grounded capacitance as follows:

$$C_{cnet_i}(t_i) = cc_{coef}(t_i) \times total\_congestion_{net_i} \tag{4}$$
$$C_{g_{net_i}}(t_i) = cg_{coef}(t_i) \times total\_length_{net_i} \tag{5}$$

To find the linear coefficients for different interconnect technology t_j, we scale the coeficients from technology t_i as follows:

$$cc_{coef}(t_j) = \frac{C_{c_{unit}}(t_j)}{C_{c_{unit}}(t_i)} cc_{coef}(t_i) \tag{6}$$
$$cg_{coef}(t_j) = \frac{C_{g_{unit}}(t_j)}{C_{g_{unit}}(t_i)} cg_{coef}(t_i) \tag{7}$$

The coupling and ground capacitances for each net in technology t_j are then computed from the congestion map obtained from estimated global routing, as follows:

$$C_{cnet_i}(t_j) = cc_{coef}(t_j) \times total\_congestion_{net_i} \tag{8}$$
$$C_{g_{net_i}}(t_j) = cg_{coef}(t_j) \times total\_length_{net_i} \tag{9}$$

Using this technique, one can estimate the total coupling capacitance and total ground capacitance of a net using the congestion

map information. Note that this corresponds to estimating a different α and total capacitance (see Eqns. (1,2)) for each net. The computed coupling and grounded capacitances are distributed equally for each segment of the congestion map that the net traverses, resulting in a disributed RC netlist for the victim net. Results for this approach are given in Section 5. One limitation of this approach is that the congestion information is taken into account for a net as a whole, and not for the individual segments of the net. The resulting RC netlist for a victim net using this technique will have the same α ratio ($\frac{C_c}{C_c+C_g}$) for all segments of the net. This issue is addressed in probabilistic approach presented in the following Section.

3.3. Probabilistic Extraction

In this method, we perform a probabilistic estimation of the coupling and grounded capacitances using the congestion information for each segment 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 other nets (Figure 4(a)). This density decreases as the nearest neighboring nets occupy farther away tracks (lower congestion) (Figure 4(b)). Figure

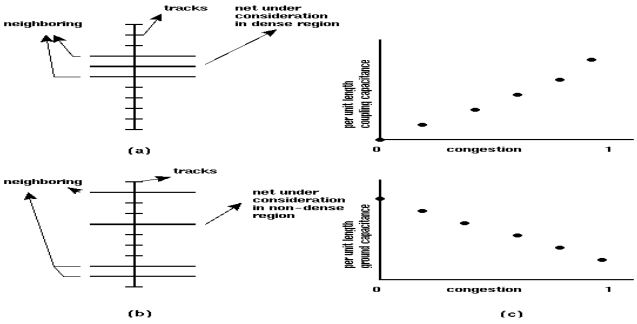

Figure 4. Dense and non-dense configurations on a congestion map segment

4(c) shows how the per unit length coupling capacitance and ground capacitance vary with different density configurations.

In our probabilistic capacitance estimation technique we estimate the per unit length capacitance values for a net segment by enumarating the possible density configurations of the net with respect to all the other nets sharing the same segment and taking a weighted average of per unit length capacitance values shown in Figure 4(c). This technique allows one to estimate both the coupling and grounded capacitances for the net section that traverses a specific segment in the congestion map. As a result, this approach provides a different α value for each net section in the congestion map. We now quantify the approach as follows. The total number of possible configurations for a congestion map segment is:

$$total\_configurations = \left(\begin{array}{c} n \\ k \end{array} \right) k! \qquad (10)$$

where n is the number of tracks in the segment and k is the number of utilized tracks. It is infeasible to enumarate 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 effected only by the location of the nearest neighboring nets.
- The effect of a neighboring net that is more than two tracks from the net under consideration is considered insignificant.

First assumption is valid since the closest neighbors act as shields to all other nets. Second assumption is in general also valid since the field lines between two nets vanish as their seperation 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 6 unique configurations, as shown in Figure 5. The bold net

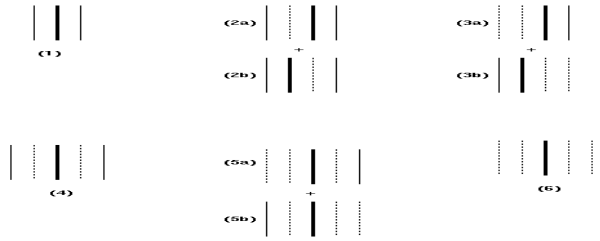

Figure 5. Configurations used

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 one 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, other side has no neighbors at all) and maximum per unit length capacitance (one side dense, other side has 2 empty tracks and then a neighbor) for these configurations.

After determining the per unit length coupling and grounded capacitance for each possible configuration, we compute probability of each configuration. The probability of each configuration depends on the number of tracks n_i, and the number of used tracks k_i. For example, the number of permutations in which the net under consideration is in configuration (1) is:

$$\begin{aligned} conf_{(1)} &= (n-2) \times \left(\begin{array}{c} k-1 \\ 2 \end{array} \right) \times 2 \\ &\quad \times \left(\begin{array}{c} n-3 \\ k-3 \end{array} \right) \times (k-3)! \end{aligned} \qquad (11)$$

We can explain Equation (11) as follows. For the victim net to be in a dense region, it can be anywhere except for the boundry

tracks of the congestion map segment under consideration. There are $(n-2)$ such tracks. There needs to be 2 nets in its neighboring tracks. These 2 nets can be chosen among the remaining $k-1$ nets and can be in any order, resulting in the second and third factor in (11). Finally, rest of the nets should be placed on the remaining tracks. There are $n-3$ nets and $k-3$ tracks left, and the nets can be in any order, resulting in the last two factors in (11). If we look at configuration (3a), the number of configurations where one side is dense and the other side has at least two empty tracks is

$$\begin{aligned} conf_{(3a)} &= (n-3)\times(k-1)\times\left(\begin{array}{c} n-4 \\ k-2 \end{array}\right)(k-2)! \\ &+ (k-1)\times\left(\begin{array}{c} n-3 \\ k-2 \end{array}\right)(k-2)! \\ &+ (k-1)\times\left(\begin{array}{c} n-2 \\ k-2 \end{array}\right)(k-2)! \end{aligned} \quad (12)$$

The second and third terms in Equation (12) are due to special cases where the victim net can be on the boundary or one track away from the boundary of the cell. The number of configurations for all other cases in Figure 5 can be calculated similarly.

For each segment that a net traverses, we compute the weighted contribution of the 6 possible configurations. Each configuration has a precomputed per unit length coupling capacitance $cc_{(i)}$, and per unit length grounded capacitance $cg_{(i)}$. We then weight the capacitance values of each configuration by the probability of the configuration, which depends on the number of tracks n_i and the track utilization k_i:

$$cc_{total(i)} = cc_{(i)} \times \frac{conf_{(i)}}{total\_configurations} \quad (13)$$

$$cg_{total(i)} = cg_{(i)} \times \frac{conf_{(i)}}{total\_configurations} \quad (14)$$

The 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 segment that a net traverses, a coupled RC circuit representation is constructed, as illustrated in Figure 1.

3.4. Aggressor Strength Estimation

We can use the information in the congestion map to estimate an average aggressor for each net. All nets that share a congestion map segment are possible neighbors. The likelihood of a net to be an aggressor to another net increases as the number of shared segments increases. To estimate an average aggressor for a net, we first find its 10 possible neighbors with the highest number of shared segments. For each of these nets, we apply the following procedure to find their signal transistion time $t_{r_{out}}$:

- Find total capacitance C_t of the net, using methods explained in previous sections.
- Obtain the Thevennin model of the driver gate using precharacterized information from the cell library.
- Compute transition time $t_{r_{out}}$ using the model shown in Figure 6

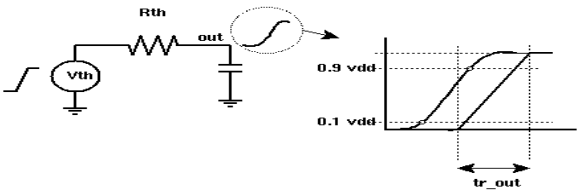

Figure 6. Model to find t_r of aggressor line

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

$$v_{out}(t) = \begin{cases} \frac{-RC+t+RCe^{-t/RC}}{t_r} & 0 < t \le t_r \\ \frac{-RC}{t_r}(e^{t_r/RC}-1)e^{-t/RC}+1 & t > t_r \end{cases} \quad (15)$$

Although it is not possible to solve Equation 15 analytically, the rise time at node *out* can easly be computed using binary search. After finding the 10% and 90% time points at node *out*, $t_{r_{out}}$ is found as shown in Figure 6. An average aggressor transition time for a victim net is then found as the weighted average (weighted by the number of shared segments) of the $t_{r_{out}}$ values for all nets considered as its possible neighbors.

4. Results

In this section, we present results on two high performance microprocessors in 0.18μ technology. Chip 1 has 58000 nets whereas chip 2 has 125000 nets. We first look at how the methods presented in this paper estimate parameters such as total coupling capacitance in the pre-route stage. Figure 7 shows the errors in the coupling capacitances (pre-route estimated vs. detailed route extracted) using the three methods described. Method 1 refers to the Steiner tree

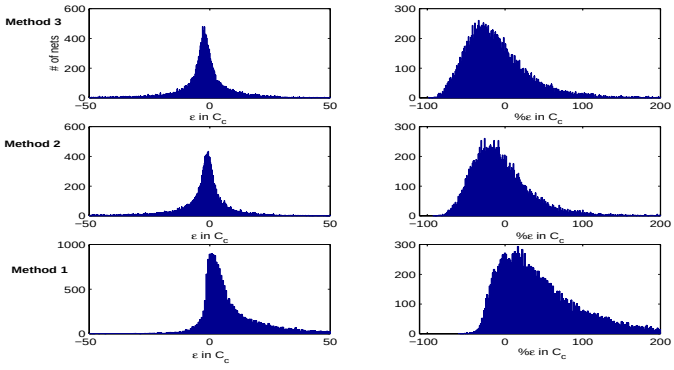

Figure 7. Error in C_c

routing based approach with an α ratio of 0.5. Method 2 and 3 refer to global routing based approaches. In method 2, the calibration method described in Section 3.2 is used and in method 3 the probabilistic extraction method described in Section 3.3 is used. First column of graphs shows the absolute errors in femtofarads whereas the second column of graphs shows the percentage errors. The average percentage error for Methods 1, 2 and 3 are 52.74%, -2% and -7.2% respectively. These errors for chip 2 are 85.14%, 16%

and 17.7%. Similar error rates are also seen for other interconnect model parameters such as total grounded capacitance C_g, α and total interconnect resistance R. As can be seen, using congestion map information provides more accurate model parameters than steiner tree routing. Figure 8 shows the absolute errors in α for chip 1 using methods 2 and 3.

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 segment that a net is going through in the congestion map whereas the calibration method assigns one α value for all net segments.

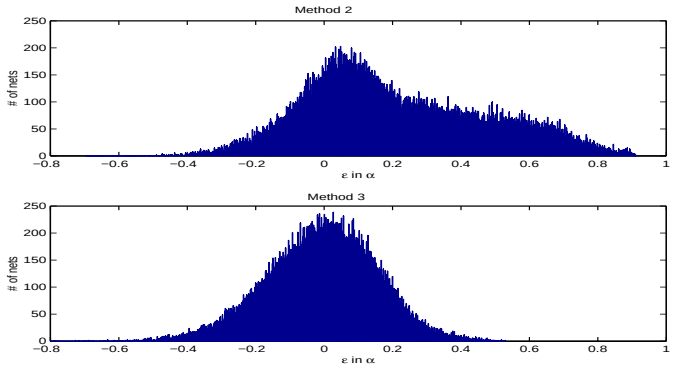

Figure 8. Error in α

Finally, in Table 1, we present the number of failing nets found after estimated noise analysis using the described method in Section 2. We compare these results with the number of failures using post-route noise analysis on the same design after detailed routing and full extraction. The column named 'common' shows the number of nets that fail in both pre- and post-route noise analysis. The column named 'missed' shows the number of nets not identified by pre-route noise analysis that subsequently failed in post route noise analysis. The column named 'false' shows the number of nets that failed in pre-route noise analysis but not in post-route noise analysis. Idealy the number of missed and false failures identified by the pre-route noise analysis is zero. False failures unnecesarily increase the required design resources to fix noise problems before routing. Missed failures, on the otherhand, are especially detrimental, since they require noise fixes after routing which can require re-routing and additional design cycles. Table 1 shows that the Steiner tree based method has dramatically more false failures than the other methods whereas the difference in number of missed nets is relatively small. Using methods 2 or 3 reduces the number of false failures by as much as 60% or 2050 nets. Also, method 3 performs better than method 2 in all aspects, resulting in a small number of missed and false violations by as much as 22% and resulting in an increased number of common nets by as much as 10%.

Chip	Method	Missed	Common	False
chip-1	Steiner	148	795	531
chip-1	Calibration	207	635	241
chip-1	Probabilistic	171	671	210
chip-2	Steiner	134	487	3509
chip-2	Calibration	193	428	1591
chip-2	Probabilistic	151	470	1459

Table 1. Failing nets in pre and post route noise analysis

5. Conclusion

In this paper, we presented pre-route noise analysis methods using estimated routing. We showed the close correlation between the detailed router and estimated global router that we used and proposed methods to extract interconnect parameters and coupling capacitances for each net using congestion information. We proposed two congestion based methods for parameter extraction: calibration method and probabilistic extraction method. We also proposed a method for aggressor strength estimation. Results show that, a good pre-route coupled extraction for each net can be made using our congestion based methods. We also showed that a pre-route noise analysis can be performed using this estimated extraction for each net, providing the designer with a high percentage of post-route problematic nets in this early design stage. The congestion map based probabilistic extraction method was shown to create least overhead (false failures). Methods presented in this paper can be used in identifying future noisy nets at a very early design stage with minimal overhead, which is a very important task in today's IC design.

References

[1] 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 *Proceedings of Design Automation Conference DAC*, pages 233–238, June 2000.

[2] S. I. Association. The international technology roadmap for semiconductors, 1999.

[3] D. Blaauw, A. Devgan, and A. Dharchoudhury. Signal integrity in high performance design. Tutorial at ICCAD, 1999.

[4] J. Cong, D. Zhingang, and P. V. Srinivas. Improved crosstalk modeling for noise constrained interconnect optimization. In *Proceedings of ASP/DAC Asia South Pasific Design Automation Conference*, pages 373–378, 2001.

[5] M. Kuhlmann and S. S. Sapatnekar. Exact and efficient crosstalk estimation. *IEEE Transactions on Computer Aided Design*, 20(7):858–866, July 2001.

[6] A. Odabasioglu, M. Celik, and L. T. Pileggi. PRIMA: Passive reduced-order interconnect macromodelling algorithm. *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, 17(8):645–653, August 1998.

[7] K. L. Shepard and V. Narayanan. Noise in deep submicron digital design. In *Proceedings of ICCAD-96 Intl. Conference on Computer Aided Design*, pages 524–531, November 1996.

[8] S. Sirichotiyakul, D. Blaauw, C. Oh, R. Levy, V. Zolotov, and J. Zuo. Driver modeling and alignment for worst-case delay noise. In *Proceedings of Design Automation Conference DAC*, pages 720–725, June 2001.

[9] A. Vittal and M. Marek-Sadowska. Crosstalk reduction for VLSI. *IEEE Transactions on Computer Aided Design*, 16:290–298, March 1997.

[10] H. Zhou and D. F. Wong. Global routing with crosstalk constraints. In *Proceedings of Design Automation Conference DAC*, pages 374–377, 1998.