Towards Global Routing With RLC Crosstalk Constraints *

James D. Z. Ma and Lei He
Department of Electrical and Computer Engineering
University of Wisconsin, Madison, WI 53706

Abstract

Conventional global routing minimizes total wire length and congestion. Experiments using large industrial benchmark circuits show that up to 24% of nets in such routing solutions may have RLC crosstalk violations at 3GHz clock. We develop an extremely efficient length-scaled K_{eff} (LSK) model that has a high fidelity for long-range RLC crosstalk. We formulate an extended global routing problem (denoted as GSINO) to consider simultaneous shield insertion and net ordering with RLC crosstalk constraints, then propose an effective three-phase GSINO algorithm. The GSINO algorithm *completely* eliminates the RLC crosstalk violations, and has small area and wire length overhead compared to conventional routing.

Categories and Subject Descriptors

B.7.2 [Design Aids] – Placement and Routing

General Terms

Algorithms, Design, Theory

1. INTRODUCTION

As VLSI technology advances, crosstalk becomes increasingly critical. Global routing has been studied to consider track assignment [1] or separated layer/track assignment and shield insertion [2] for capacitive crosstalk constraints. A few recent works have addressed crosstalk avoidance techniques for both capacitive and inductive crosstalk. Examples include shielding [3], simultaneous shield insertion and net ordering (SINO) [4], twisted bundle layout structure [5], and differential signaling [6]. However, there has been no in-depth study on automatic global routing that is able to consider both capacitive and inductive crosstalk constraints.

The rest of the paper is organized as follows: Section 2 introduces the preliminaries and proposes a length-scaled K_{eff} (LSK) model. Section 3 formulates an extended global routing ($GSINO$) problem and develops a three-phase algorithm. Section 4 presents the experimental results. Section 5 concludes the paper with discussions on future work. A full version of this paper is available as a technical report [7], including details of LSK model development and verification, in-depth GSINO algorithm description, and more experimental results.

*This research is partially supported by NSF CAREER Award CCR-0093273 and Intel. We used computers donated by HP and SUN Microsystems. Lei He is now Assistant Professor of Electrical Engineering Department at UCLA. Address comments to lhe@ee.ucla.edu.

DAC 2002, June 10-14, 2002, New Orleans, Louisiana, USA.

2. RLC NOISE MODELING

2.1 Preliminaries

We consider over-the-cell routing for global interconnects in this paper. We denote a set of signal nets as $\mathcal{N} = \{N_1, N_2, \cdots\}$. Each net N_i has a number of pins $(p_{i0}, p_{i1}, \cdots)$, where p_{i0} is the source and p_{ij} $(j > 0)$ is the sink. We assume a pair of routing layers, one for horizontal wires and the other for vertical wires. The routing layers are divided by pre-routed power/ground (P/G) networks into routing *regions* $\mathcal{R} = \{R_1, R_2, \cdots\}$. The number of horizontal (vertical) tracks available for region R_k is the horizontal (vertical) capacity $HC(R_k)$ ($VC(R_k)$). A track can be occupied by a segment of either a net or a shield. A shield is a wire directly connected (without through devices) to P/G networks. One convenient way to connect shields is to add vias between shields and P/G networks. We also assume that all global interconnects have the same driver resistance and loading capacitance, and all wires (except P/G wires) have the same width, spacing, and thickness. Moreover, P/G wires are wide enough so that there is *no* crosstalk (coupling) between regions separated by P/G wires.

According to [4], two signal nets N_1 and N_2 are *sensitive* to each other if a switching event on N_1 causes N_2 to malfunction (due to extraordinary crosstalk or delay variation). In this case we call N_1 an aggressor for N_2 and N_2 a victim of N_1. The *sensitivity rate* of N_i is defined as the ratio of the number of aggressors for N_i to the total number of signal nets. Further, we assume that there is no coupling between different routing regions. The simultaneous shield insertion and net ordering (SINO) problem [4] has been studied to find the minimum area solution with the following RLC crosstalk constraints: all signal nets are capacitive crosstalk free (i.e., no sensitive nets are adjacent to each other) and have inductive crosstalk (or equivalently, the total crosstalk) less

than a given threshold. An accurate and extremely efficient model for long-range RLC crosstalk, such as the LSK model to be developed below, is of paramount importance for a routing algorithm considering SINO at the full chip level.

2.2 LSK Model

A formula-based K_{eff} model has been proposed in [4] to characterize the inductive coupling between two signal nets. It uses a formula to calculate the coupling coefficient K_{ij} between two signal nets N_i and N_j. The total amount of inductive coupling K_i induced on N_i is $\sum_{j \neq i} K_{ij}$ for all signal nets that are sensitive to N_i. The K_{eff} model is easy to compute and convenient to use at a higher design level or an earlier design stage. Furthermore, it has a high fidelity in the following sense [8]: for a SINO solution of multiple nets with a fixed wire length, a signal net with a higher K_i value given by the K_{eff} model also has a higher SPICE-computed noise voltage. Moreover, empirical evidence [7] has shown that the noise voltage is roughly a linearly increasing function of the wire length and this observation holds for a large number of SINO solutions in our experiments considering different design and fabrication technologies.

Therefore we propose the following length-scaled K_{eff} model (in short, LSK model). We first compute the LSK value as

$$LSK \quad = \quad \sum_j l_j \cdot K_i^j \qquad (1)$$

where K_i^j is the total inductive coupling for the net N_i under study in region R_j and l_j is the length of N_i in R_j. We then compute the RLC crosstalk voltage from the LSK value by looking up a table with two columns, one for LSK and the other for the corresponding crosstalk voltage. To build such a table, we generate a number of SINO solutions for a single routing region, and compute the LSK values and corresponding crosstalk voltages via SPICE simulations for different wire lengths. Our table used in the paper contains 100 entries, with crosstalk voltage values from 0.10V to 0.20V, which is about 10% $\sim$ 20% of the supply voltage Vdd (1.05V for the ITRS 0.10μm technology [9]).

Note that the LSK model can only be applied to a net with the following property: none of the neighboring wires of N_i switches simultaneously with N_i. All nets in a SINO solution satisfy this property. We also assume uniform driver and receiver for all interconnects, and the aforementioned table should be re-computed for different combinations of driver and receiver. Our future work includes generalizing the LSK model for non-uniform drivers and receivers

3. GSINO PROBLEM FORMULATION AND ALGORITHM

We formulate the GSINO problem as follows:

FORMULATION 1. *The optimal GSINO problem decides a Rectilinear Steiner Tree RST_i for each net N_i in the context of global routing, and finds a SINO solution within each region such that the given RLC crosstalk constraint is satisfied for each sink, and the total wire length and routing area are minimized.*

The GSINO problem has a high complexity, as even its sub-problem SINO is NP-hard [4]. Therefore, we propose the following three-phase heuristic algorithm. In Phase I, we *uniformly* partition the crosstalk bounds among routing regions for each signal net, and perform global routing with consideration of allocating and minimizing shielding area. In Phase II, we find a SINO solution within each routing region under partitioned crosstalk bounds. In Phase III, we carry out local refinement to eliminate the remaining (but very limited) crosstalk violations and further reduce the routing congestion. We only present details of Phases I and III below, and refer readers to [4] for the SINO algorithm used in Phase II.

3.1 Phase I Algorithm

For simplicity of presentation, we assume a uniform constraint of crosstalk voltage for all sinks. Both our algorithm and program implementation, however, can handle non-uniform crosstalk constraints. For crosstalk bound partitioning we first map the crosstalk voltage into an LSK value by the table look-up approach described in Section 2. We then use $L_{e,ij}$, the Manhattan distance between the source p_{i0} and a specific sink p_{ij} of net N_i, to approximate the wire length in the final routing solution, and compute the inductive coupling bound for each net segment on the path from the source to the sink as $K_{th} = \frac{LSK}{L_{e,ij}}$. For the net segment on the common paths from the source to multiple sinks, K_{th} is the minimum of those bounds determined for individual paths.

After the above uniform partitioning of the crosstalk bounds, K_{th} is available and fixed for each net segment in global routing phase. We then apply the iterative deletion (ID) algorithm [10] to synthesize a global routing solution with shielding area (i.e., a number of tracks) properly reserved and more importantly minimized. We choose the ID algorithm because it is able to consider all the signal nets simultaneously and thus is independent of the net routing order. It is less efficient but may lead to better solutions compared to other order-dependent routing approaches. Nevertheless, our algorithm framework can be applied to other routing algorithms.

We first define the *net connection graph* for net N_i as an undirected graph $G_i = (V_i, E_i)$, where V_i is the set of regions within the bounding box for the pins of net N_i and there is an edge $e_{i,jk} \in E_i$ if $V_{i,j}$ and $V_{i,k}$ are adjacent regions. Let Ω be the set of the connection graphs we construct for all the nets. We iteratively delete the edge with the largest weight from Ω until Ω is a net connection forest, i.e., the connection graphs for all nets are reduced to connection trees. The ID algorithm is summarized in Figure 1, with details in [10] and [7].

ID Algorithm
For each net N_i, construct G_i
Let Ω be the set of G_i's.
Repeat
    e = the edge with the maximum weight in Ω.
    remove e from the corresponding G_i.
    update the weight of affected edges.
Until Ω is a net connection forest.

Figure 1: ID algorithm.

The weight of a horizontal edge is computed as

$$w(e) \quad = \quad \alpha \cdot f(WL) + \beta \cdot HD(R_i) + \gamma \cdot HOFR(R_i) \qquad (2)$$

where $f(WL)$ is the normalized wire length with respect to

the estimated wire length of the Rectilinear Steiner Minimum Tree ($RSMT$) for the current net. $HD(R_i)$ is the routing density defined as $\frac{HU(R_i)}{HC(R_i)}$, where $HC(R_i)$ is the horizontal capacity of region R_i and $HU(R_i) = N_{ns} + N_{ss}$ is the utilization of horizontal tracks, with N_{ns} being the number of net segments in the region and N_{ss} being the number of shields needed by the min-area SINO solution to satisfy the K_{th} constraint for each net segment in the region. Moreover, $HOFR(R_i)$ is the relative horizontal overflow, i.e., the number of overflow net segments over the routing capacity. Finally, α, β, and γ are three constants and can be tuned for different objectives. Generally, γ is much larger than α and β so that virtually no overflow is allowed in the final global routing solution. We set α=2, β=1, and γ=50 in this paper. The weight for a vertical edge can be defined by a formula similar to Formula 2.

Note that given the fixed K_{th} for each net segment, the number of shields in each region is a function of the number of net segments N_{ns} in the region and their sensitivities (S_i's) [8, 7]. I.e.,

$$N_{ss} = a_1 \cdot \sum_{i=1}^{N_{ns}} S_i^2 + a_2 \cdot \frac{1}{N_{ns}} \cdot \sum_{i=1}^{N_{ns}} S_i^2 + a_3 \cdot \sum_{i=1}^{N_{ns}} S_i + a_4 \cdot \frac{1}{N_{ns}} \cdot \sum_{i=1}^{N_{ns}} S_i + a_5 \cdot N_{ns} + a_6 \quad (3)$$

In [7] we present the values of the coefficients and verify the formula using a large range of N_{ns} and S_i. The estimates differ by at most 10% from the min-area SINO solutions.

It is worthwhile to emphasize that according to Formula (3), the number of shields in a region is proportional to the density of sensitive nets in that region. Because the net segments in different regions are not sensitive to each other by definition and the weight of an edge considers the density of sensitive nets, our new ID algorithm is able to evenly distribute the sensitive nets across the chip, avoid routing too many sensitive nets in the same region, and reduce the total number of shields.

3.2 Phase III Algorithm

During the crosstalk budgeting in Phase I, we use the Manhattan distance from source to sink to approximate the wire length of the final route. A detour may actually occur, cause under-estimation of the crosstalk, and lead to crosstalk violations. Even though such violations are very limited in our experiments, we use the algorithm shown in Figure 2 to *completely* eliminate the crosstalk violations and further reduce the routing congestion. Phase III contains two passes of iterative local refinement (denoted as LR) in a greedy fashion, considering crosstalk bound re-distribution which is different from the uniform distribution used in Phase I.

There are two loops in the first pass to eliminate the crosstalk violations. The outer loop picks the net N_i with the most severe crosstalk violation, and the inner loop locates the least congested routing region R_j through which N_i is routed. Further, the inner loop allows exactly *one* more shield to be added into R_j by using Formula (3) to decide how much the K_{th} can be reduced for the net segment of N_i. Finally, SINO is invoked again in R_j with respect to the reduced K_{th}. This inner loop is iterated until we eliminate the crosstalk violation for N_i. The outer loop is stopped when there are no more crosstalk-violating nets.

There are also two loops in the second pass to reduce the congestion. In the outer loop, we start with the most congested region R_j and compute the *slack* as K_i - K_{th} for all the nets that are routed through region R_j. To avoid being too greedy, in the inner looper we allow only *one* shield to be removed at a time. Again, Formula (3) is used to decide how much increase on K_{th}'s is needed. The inner loop iteration is stopped when one shield can be removed. We then re-run SINO under the increased K_{th}'s. We only accept the new SINO solution if it has no crosstalk violations, and in this case update the congestion of region R_j and the slacks of affected nets. The outer loop is stopped when no reduction on slacks is possible without causing crosstalk violations.

```
Pass 1: Eliminate crosstalk violations
Repeat
    N_i = net with most severe crosstalk violation
    Repeat
        R_j = the least congested region containing N_i
        decrease K_th for N_i's segment by
        allowing one more shield in R_j
        re-do SINO in R_j
    Until N_i has no crosstalk violation
Until no nets have crosstalk violations

Pass 2: Reduce routing congestion
Repeat
    R_j = the most congested region
    compute the slacks for nets routed through region R_j
    Repeat
        N_i = the net with the largest slack
        increase K_th for N_i's segment in R_j by N_i's slack
    Until one shield can be removed from R_j
    new_SINO = re-do SINO in R_j
    if (new_SINO causes no crosstalk violations) then
        SINO=new_SINO
        update the congestion of region R_j
        update the slacks of affected nets.
Until no reduction on the slacks is possible without
        causing crosstalk violations
```

Figure 2: LR algorithm.

4. EXPERIMENTAL RESULTS AND DISCUSSIONS

We have implemented the GSINO algorithm using C/C++ on a UNIX workstation, and applied it to the ISPD'98/IBM benchmark circuits at 3GHZ clock (see [7] for detailed description of the circuits and the derived technology parameters). The placement is generated by DRAGON [11] and the crosstalk constraint is set to 0.15V, around 15% of the supply voltage Vdd, for all sinks.

We compare GSINO with the following two approaches: (1) ID+NO: ID-based global router to minimize wire length and congestion only, followed by net ordering (NO) within each region to eliminate as much capacitive coupling as possible; and (2) *i*SINO: ID-based global router to minimize wire length and congestion only, followed by SINO within each routing region. In order to make fair comparisons, we still consider the weight function (Formula (2)) but without N_{ss} in $HU(R_i)$ for both ID+NO and *i*SINO (i.e., no shielding area reservation or minimization is considered).

Both the *i*SINO and GSINO approaches apply SINO to meet the crosstalk bounds so that no crosstalk violations are observed. In Table 1 we only present the numbers of

crosstalk-violating nets in ID+NO solutions. Since ID+NO is not aware of an RLC crosstalk constraint during global routing, up to 24% of nets may have crosstalk violations.

	sensitivity rate = 30%	sensitivity rate = 50%
ibm01	1907 (14.60%)	2583 (19.78%)
ibm02	3254 (16.87%)	4275 (22.16%)
ibm03	4920 (18.85%)	6056 (23.20%)
ibm04	5143 (16.42%)	5928 (18.92%)
ibm05	4361 (14.71%)	7135 (24.07%)
ibm06	4802 (13.96%)	6573 (19.11%)

Table 1: Numbers of crosstalk-violating nets for ID+NO solutions. The data in the parentheses are the percentages with respect to the total numbers of signal nets.

Because applying SINO within each region after global routing does not change the wire length, *i*SINO has the same wire length as ID+NO. On the other hand, GSINO has an average of 7% wire length overhead compared with ID+NO for 30% sensitivity rate and 13% overhead for 50% sensitivity rate (see Table 2). It is worthwhile to point out that the SINO solution has a relatively smaller delay per unit length as no neighboring wires switch simultaneously [12]. Therefore, the performance penalty due to the increase on wire length should be less than the wire length penalty.

	ID+NO	GSINO	ID+NO	GSINO
	sensitivity rate = 30%		sensitivity rate = 50%	
ibm01	639	683 (6.89%)	639	706 (10.49%)
ibm02	724	796 (9.94%)	724	829 (14.50%)
ibm03	647	717 (10.82%)	647	753 (16.38%)
ibm04	748	815 (8.96%)	748	868 (16.04%)
ibm05	695	741 (6.62%)	695	784 (12.81%)
ibm06	769	827 (7.54%)	769	860 (11.83%)

Table 2: Average wire lengths (μm) of ID+NO and GSINO solutions. The data in the parentheses are the average increase on wire lengths compared to ID+NO solutions.

In Table 3, we calculate the routing area by the product of the maximum row and column lengths, and compare the three routing approaches. *i*SINO has large routing area overhead compared to ID+NO, 18% on average for 30% sensitivity and 23% on average for 50% sensitivity. GSINO reduces the area overhead to 7% and 9%, respectively.

	ID+NO	*i*SINO	GSINO
	sensitivity rate = 30%		
ibm01	1533 × 1824	1658 × 1974 (17.04%)	1589 × 1866 (6.04%)
ibm02	3004 × 3995	3306 × 4283 (17.99%)	3105 × 4087 (5.74%)
ibm03	3178 × 3852	3492 × 4108 (17.18%)	3285 × 3950 (6.00%)
ibm04	3861 × 3910	4157 × 4241 (16.78%)	4120 × 3932 (7.31%)
ibm05	9837 × 7286	11529 × 7443 (19.73%)	10526 × 7404 (8.74%)
ibm06	5002 × 3795	5412 × 4107 (17.09%)	5283 × 3890 (8.26%)
	sensitivity rate = 50%		
ibm01	1533 × 1824	1714 × 2048 (25.53%)	1591 × 1872 (6.51%)
ibm02	3004 × 3995	3476 × 4329 (25.39%)	3187 × 4125 (9.54%)
ibm03	3178 × 3852	3662 × 4139 (23.82%)	3318 × 4050 (9.77%)
ibm04	3861 × 3910	4278 × 4322 (22.47%)	4115 × 3951 (7.67%)
ibm05	9837 × 7286	11772 × 7489 (23.00%)	10428 × 7406 (7.75%)
ibm06	5002 × 3795	5527 × 4206 (22.46%)	5418 × 3889 (11.00%)

Table 3: Routing areas ($\mu m \times \mu m$) of ID+NO, *i*SINO, and GSINO solutions. The data in the parentheses are the increase on routing areas compared to ID+NO solutions.

In Tables 1–3, we consider two sensitivity rates 30% and 50%. In the case of 30%, a signal net is sensitive to random 30% of other signal nets in the netlist. We observe that when the sensitivity rate decreases from 50% to 30%, GSINO results in reduced wire length (26% on average) and reduced routing area overhead (20% on average). The actual sensitivity rate in real designs depends on logic and physical implementation, and hence our assumption on random sensitivity rate is only an approximation. However, we expect the overall sensitivity rate in real designs to be less than 50%, implying that the wire length and routing area overhead will be less than what we report in this work.

5. CONCLUSIONS AND FUTURE WORK

In this paper we have formulated a GSINO problem and developed an effective three-phase algorithm to satisfy the RLC crosstalk constraints, with consideration of SINO. Experimental results have demonstrated that the GSINO algorithm is able to completely eliminate the RLC crosstalk violations, with small routing area and wire length overhead.

The majority of running time in the current three-phase GSINO algorithm is consumed by the ID-based global routing phase. A more efficient global router will be developed or be integrated into the GSINO framework. Furthermore, we plan to explore alternative crosstalk budgeting approaches.

6. ACKNOWLEDGMENT

The authors would like to thank Dr. Prashant Saxena at Intel, and Mr. Kevin Lepak and Mr. Jinjun Xiong at University of Wisconsin-Madison for helpful discussions.

7. REFERENCES

[1] T. Xue and E. S. Kuh, "Post global routing crosstalk synthesis," *TCAD*, pp. 1418–1430, Dec. 1997.

[2] H. Zhou and D. F. Wong, "Global routing with crosstalk constraints," *TCAD*, November 1999.

[3] L. He, N. Chang, S. Lin, and O. S. Nakagawa, "An efficient inductance modeling for on-chip interconnects," in *CICC*, 1999.

[4] L. He and K. M. Lepak, "Simultaneous shielding insertion and net ordering for capacitive and inductive coupling minimization," in *ISPD*, 2000.

[5] G. Zhong, H. Wang, C.-K. Koh, and K. Roy, "A twisted bundle layout structure for minimizing inductive coupling noise," in *ICCAD*, 2000.

[6] Y. Massoud, J. Kawa, D. MacMillen, and J. White, "Modeling and analysis of differential signaling for minimizing inductive cross-talk," in *DAC*, 2001.

[7] J. D. Ma and L. He, "Towards global routing with RLC crosstalk constraints," in *University of Wisconsin, Technical Report, ECE-02-1*, 2002.

[8] J. D. Ma, A. Parihar, and L. He, "Pre-routing estimation of shielding for RLC signal integrity," in *ICCD*, 2001.

[9] Semiconductor Industry Association, *International Technology Roadmap for Semiconductors*, 1999.

[10] J. Cong and B. Preas, "A new algorithm for standard cell global routing," *Integration, the VLSI Journal*, November 1992.

[11] M. Wang, X. Yang, and M. Sarrafzadeh, "DRAGON2000: Standard-cell placement tool for large industry circuits," in *ICCAD*, 2000.

[12] J. D. Ma and L. He, "Formulae and application of interconnect estimation considering shield insertion and net ordering," in *ICCAD*, 2001.