

University of Wisconsin
Electrical and Computer Engineering Department
Technical Report ECE-00-6

Simultaneous Shield Insertion and Net Ordering
for Coupled RLC Nets under Explicit Noise Constraint

Kevin M. Lepak, Irwan Luwandi, and Lei He *
Electrical and Computer Engineering Department
University of Wisconsin-Madison, WI 53706

Abstract

For a number of coupled RLC nets, we formulate the min-area simultaneous shield insertion and net ordering (SINO/NB- v) problem to satisfy the given noise voltage bound. We develop an efficient model to compute the noise voltage, and apply the noise model to a simulated-annealing (SA) based algorithm for the SINO/NB- v problem. Extensive and detailed experiments show that the SA-based algorithm is efficient, and always achieves solutions satisfying the given noise bound. It uses up to 249% and 37% fewer shields when compared to a greedy based shield insertion algorithm and a separated shield insertion and net ordering algorithm, respectively. To the best of our knowledge, it is the first work that presents an in-depth study on min-area SINO problem for multiple RLC nets under an explicit noise constraint.

*This research was partially supported by a grant from Intel, and used computers donated by SUN Microsystems. Mr. Lepak is supported by an Intel fellowship. Mr. Luwandi's part in this research was performed while he was with University of Wisconsin. He now works with Intel. Address comments to lhe@ece.wisc.edu.

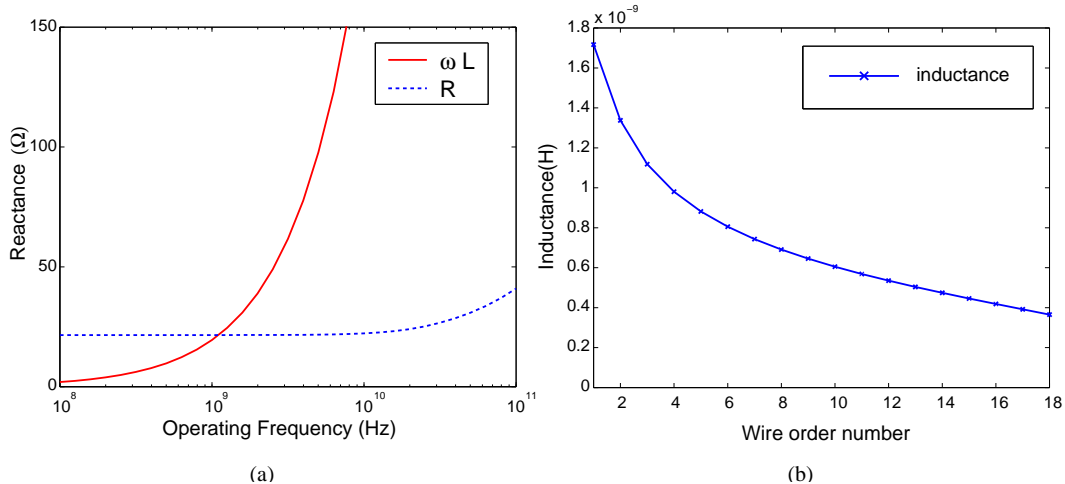


Figure 1: (a) Characteristics of resistance and reactance due to inductance; (b) Characteristics of self inductance and mutual inductance.

1 Introduction

Given the growing importance of interconnects in performance, reliability, cost, and power dissipation for future high-performance and power-efficient circuits and systems, an *interconnect-centric* design methodology is clearly essential for analyzing and optimizing interconnects at every design stage [1, 2]. Interconnect synthesis is a critical aspect of interconnect-centric design, and has been extensively studied for RC interconnect models. Existing work under RC models includes routing topology generation, wire sizing, wire spacing, device sizing, buffer insertion, net ordering, and combinations of these design components. A comprehensive survey on interconnect synthesis under RC interconnect models can be found in [3].

However, RC interconnect models become increasingly inadequate as the on-chip inductive effect gains prominence in gigahertz designs. A simple rule of thumb is that the inductance should be considered if resistance R and reactance ωL have similar values, where L is inductance and $\omega = 2\pi f$ with f being the operating frequency. In Figure 1(a), we compare R and ωL under different operating frequencies. We used the three-dimensional electromagnetic field solver FastHenry [4] to compute R and ωL for a typical global interconnect, which is $0.8\mu\text{m}$ wide, $2\mu\text{m}$ tall, and $2000\mu\text{m}$ long. One may easily see that ωL starts to outweigh the resistance at the operating frequency of approximately one gigahertz. As the operating frequency is larger than the clock frequency due to the harmonic effect,¹ on-chip inductance should be considered in the layout design for circuits of gigahertz clock frequencies.

Furthermore, we compare the self inductance and mutual inductance in Figure 1(b) above. We designed an eighteen-bit signal bus sandwiched between two coplanar power/ground nets, where all wires are $0.8\mu\text{m}$ wide, $2\mu\text{m}$ tall, and $2000\mu\text{m}$ long, and are separated by $0.8\mu\text{m}$. We computed the loop inductance for these wires using FastHenry. In Figure 1(b), the leftmost data-point stands for the self inductance of the leftmost signal net, and the rest of the data-points are mutual inductance between the leftmost signal net and the remainder

¹The operating frequency is decided by the signal rise time t_r . The knee frequency can be defined as $F_{knee} = 0.5/t_r$ and be used as the operating frequency to compare ωL and R , as “the behavior of a circuit at (operating) frequencies above F_{knee} hardly affects digital performance” [5]. A similar conclusion was drawn in [6] using the concept of “significant frequency”. More sophisticated rules to judge the significance of inductance can be found in [7, 8, 9].

of the signal nets. Clearly, the inductance has a “long-range” effect in the sense that the mutual inductance between non-adjacent nets cannot be ignored when compared to the self inductance. Note that the capacitance is a “short-range” effect in the sense that the mutual capacitance between non-adjacent nets is negligible, so that interconnect synthesis under the RC model needs to consider only the net under study and its two adjacent nets, or in most cases, to consider just the net under study by assuming the worst-case mutual capacitance to the adjacent nets. This single-net based approach no longer works for the RLC model due to the long-range inductive effect. Hence, interconnect synthesis under the RLC model should consider multiple coupled nets simultaneously, and is *inherently* more difficult compared to interconnect synthesis under the RC model.

There has been some preliminary work using the RLC interconnect models, including routing topology generation [10], wire sizing [11] and buffer insertion [12]. However, all of them assume a single RLC net, which is not adequate as discussed above. One very recent work considers interconnect synthesis for multiple coupled nets [13]. It is shown that the existing net order formulation in [14, 15, 16, 17, 18, 19] is no longer effective for minimizing noise with the presence of on-chip inductance, and that a shield which is a wire directly connected to P/G nets is able to reduce the inductive noise. Then, the simultaneous shield insertion and net ordering (SINO) problem is formulated for multiple RLC nets, finding a minimum-area solution that is free of capacitive noise and satisfies the given bound of inductive noise.² The problem is NP-hard, but an approximate algorithm based on simulated annealing (SA) is developed, achieving satisfactory solutions in experiments.

Yet, the inductive coupling coefficient is used as a figure of merit for the inductive coupling in [13], without dealing with noise voltage at all. Our contributions in this paper include: (i) We develop an efficient model to compute the peak noise voltage, considering both capacitive coupling and “long-range” inductive coupling. (ii) Applying the new noise model, we formulate and solve a new SINO problem (herein refer to as the SINO/NB- v problem) to find a min-area SINO solution for the given noise constraint. (iii) We verify our SINO solution using SPICE simulation and an accurate RLC circuit model based on the partial inductance model. Experiments show that our SINO solutions always satisfy the given noise bound, and use up to 249% and 37% fewer shields when compared to a greedy based shield insertion algorithm and a separated shield insertion and net ordering algorithm. Further, the algorithm is efficient to optimize a 32-bit bus in approximately ten seconds. To the best of our knowledge, it is the first work that presents an in-depth study on the min-area simultaneous shield insertion and net ordering problem for multiple RLC nets under an explicit noise constraint.

The rest of this paper is organized as follows: Section 2 reviews the related work and formulates the new SINO problem. Section 3 presents an efficient noise voltage model considering both capacitive and inductive coupling. Section 4 introduces algorithms to solve the SINO/NB- v problem. Section 5 describes the tool implementation, analyzes experimental results, and discusses the implications to future multi-GHz designs. Section 6 concludes the paper, with discussion of future work.

²Shields pre-routed with regular patterns have been used in [20, 21] for noise minimization (the dense wire fabric approach in [21] also leads to a much reduced coupling capacitance and much reduced delay variations). However, the two approaches do not leverage quiet wires that may serve as the current return path. They may lead to over-shielding and an increased routing area when compared to the SINO approach.

2 Problem Formulations

2.1 Preliminaries

2.1.1 Coplanar Interconnect Structures

Throughout this work, we consider only parallel coplanar interconnect structures with all wires having the same length (extensions to more general interconnect structures will be discussed in Section 6). These are characterized as a number of signal traces and power/ground traces which run parallel in the same layer. We give an example of this structure in Figure 1. In the Figure, P and G represent the power and ground grids (P/G grids), s represents signal wires (denoted as *s-wires*), and g is a shield wire (in short, a *shield*) that often has similar width as an s-wire and is connected to P/G grids. Both P/G grids and shield wires provide dedicated current return paths for signals, and are denoted as *g-wires* in this paper. We will use the terms “wire” and “net”, as well as “shield” and “g-wire” interchangeably.

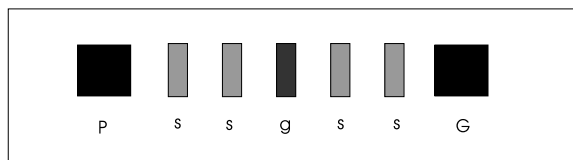


Figure 2: A cross-section view of a coplanar interconnect structure with a shield inserted.

An interconnect structure can be represented by a string, where each symbol stands for an s-wire or a g-wire. For example, the interconnect structure in Figure 2 can be represented by $gssgssg$ if we do not distinguish these s-wires (or alternatively, $g2sg2sg$). If we label the s-wires from left to right as s_1, s_2, s_3 and s_4 , then the string $gs_1s_2gs_3s_4g$ is a unique representation of a *shield insertion and net ordering* solution (referred to as a *SINO* solution or a SINO string). In this paper, a SINO string implicitly includes a shield trace (the power and ground grids) as its first and last element. These P/G grids are also shield resources. As an example, consider the following: $\langle g \rangle s_1s_3gs_2s_4s_5gs_7s_6s_0 \langle g \rangle$. This string represents an eight (signal) bit interconnect structure with two g-wires (plus two implicit g-wires for the P/G grids denoted as $\langle g \rangle$). Note that we can apply our formulations and algorithms to be presented to any group of wires which may contain pre-routed g-wires more than just a pair of P/G grids. We call the group of wires sandwiched between adjacent g-wires a *block*, and the number of s-wires in a block as the block size. A block can be represented as a substring of a SINO string (i.e. block 0 of the above string would be written as s_1s_3). In the SINO string, the g-wires on each end are implicit with the substring.

2.1.2 Net Sensitivity

We define two nets s_1 and s_2 to be sensitive to each other if a switching signal on s_1 will cause s_2 to malfunction (due to extraordinary crosstalk or delay variation) or vice-versa. As illustrated in Figure 3, the *aggressor* may cause noise in both *victim₁* and *victim₂*. For *victim₁*, the noise pulse occurs during a sampling window—hence the aggressor and *victim₁* are sensitive. For *victim₂*, the noise pulse does not occur during a sampling window—hence the aggressor and *victim₂* are not sensitive and *victim₂* still works correctly even though there is coupling between the *aggressor* and *victim₂*. Net sensitivity has been leveraged in [14, 15, 16, 17, 18, 19] for

noise minimization, by not placing sensitive nets adjacent to one another, but allowing non-sensitive nets to be placed adjacent to one another.

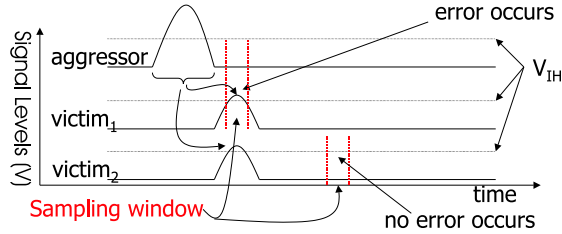


Figure 3: Illustration of net sensitivity. The y-axis indicates signal (voltage) level and the x-axis indicates time. The switching event on the aggressor induces a noise voltage in the two victims as shown. For *victim₁*, the noise pulse occurs during a sampling window—hence the aggressor and *victim₁* are sensitive. For *victim₂*, the noise pulse does not occur during a sampling window—hence the aggressor and *victim₂* are not sensitive.

The sensitivity for all s-wires in a given problem can be represented compactly with a sensitivity matrix S of size $n \times n$ (where n is the number of s-wires) as shown in Figure 4. The graphical representation of the sensitivity matrix (essentially an undirected graph structure), is also shown in Figure 4. An entry of 1,0 in location (i, j) indicates that s_i and s_j are sensitive or not sensitive, respectively, to one another. By definition, the matrix must be symmetric since the underlying graph is undirected (i.e. $S_{ij} = S_{ji}$). In this paper, we assume that an appropriate sensitivity matrix indicating design parameters and net relationship semantics is given *a priori*. By definition, a shield is not sensitive to any s-wire.

In the rest of this paper, we use the following “strict” definition for the aggressor: s-wire s_i is an aggressor for s-wire s_j if and only if the two wires are sensitive to each other, i.e., $S_{ij} = 1$. For a SINO string $s_1s_2gs_3s_4$, if s_1 and s_3 are sensitive to victim s_4 , but s_2 is not sensitive to victim s_4 , the given SINO string can be also represented as *aqgav*, meaning that for noise computation, s_1 and s_3 are aggressors for the victim and s_2 as well as the shield are quiet wires for the victim.

2.2 SINO Problems for Capacitive and Inductive Noise Minimization

2.2.1 Previous Work

Simultaneous shield insertion and net ordering (*SINO*) has been studied in [13] for capacitive and inductive noise minimization. The inductive coupling coefficient is used as a figure of merit for inductive coupling, and a K_{eff} model is developed for the inductive coupling coefficient. According to the K_{eff} model, an s-wire is inductive noise free if it does not share a block with any aggressor. Further, s_i is capacitive noise free if it is not directly adjacent to any aggressor. A placement P (or equivalently, a SINO solution, or a SINO string) is noise free if, and only if, all nets s_i within P are free of both capacitive noise and inductive noise. With respect to these concepts, the following SINO problem is defined in [13] and is called the noise free SINO problem under the K_{eff} model in this paper (in short, the SINO/NF- k problem).

Formulation 1 (*Optimal SINO/NF- k problem*) For a given placement P , find a new placement P' by simultaneous shield insertion and net re-ordering such that P' is noise free under the K_{eff} model and the total area of P' is minimal.

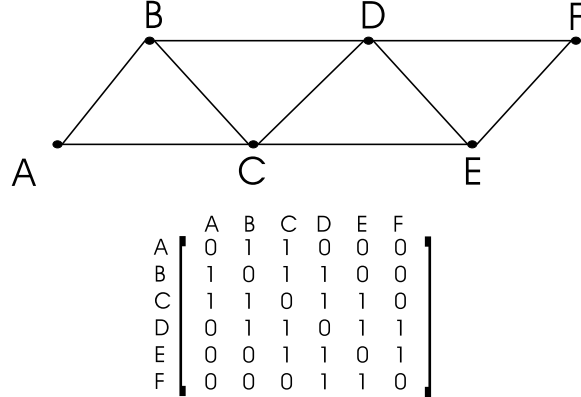


Figure 4: Illustration of a sensitivity graph and the corresponding sensitivity matrix. The nodes in the graph correspond to signal nets, with an arc between the nodes indicating that two nets are sensitive to one-another. The sensitivity matrix is an equivalent representation for the sensitivity graph shown.

Because the SINO/NF- k problem is over-constrained and may lead to over-designed solutions, the following SINO problem to meet a given noise bound is defined in [13] and is called the noise bounded SINO problem under the K_{eff} model in this paper (in short, the SINO/NB- k problem).

Formulation 2 (*Optimal SINO/NB- k problem*) For a given placement P , find a new placement P' with minimal area by simultaneous shield insertion and net re-ordering such that any s_i in P' is free of capacitive noise and its inductive coupling is less than a given bound.

The main limitation of the above two SINO problems is the K_{eff} model, which assumes that all current returns via the nearest shields. This assumption does *not* hold in general, as the current may return from quiet wires closer than the nearest shields, and the current may return from quiet wires or shields beyond the nearest shields when multiple wires within a block switch simultaneously (see discussion in Section 3.3.2). Further, the noise bound in the SINO/NB- k is not an explicit noise voltage that is most useful to the designer. Additionally, the above two SINO problems do not allow placing a victim directly adjacent to an aggressor, and may lead to over design in practice. Nevertheless, [13] was the first paper to consider automatic noise minimization with consideration of both capacitive and inductive coupling for multiple nets, and is able to obtain SINO solutions that show a smooth tradeoff between the number of shields and SPICE-computed noise voltage³.

2.2.2 New Problem Formulation

To overcome the limits in the SINO/NF- k and SINO/NB- k problem formulations, we formulate the following new SINO/NB- v problem:

Formulation 3 (*Optimal SINO/NB- v problem*): For a given placement P , find a new placement P' with the minimum area by simultaneous shield insertion and net re-ordering such that the peak noise of any wire s_i in P' satisfies the given explicit noise constraint for wire s_i .

³Private communication with authors of [13]

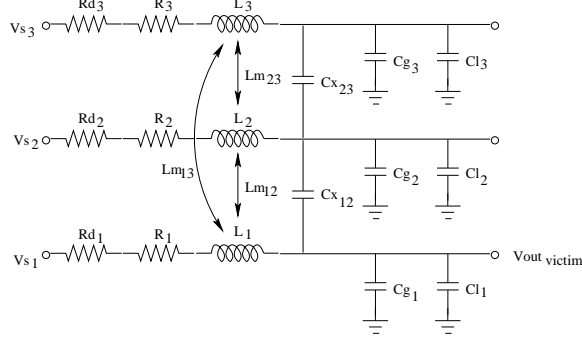


Figure 5: RLC Circuit model for three coupled wires. One RLC segment is used for each wire in this example.

In this problem formulation, the *peak noise* is the maximum noise that can be induced for the victim over all signal patterns for its aggressors. An efficient model for the peak noise is of paramount importance to solving the SINO/NB- v problem. We will present such a noise model in Section 3. Further, our noise model, SINO algorithms and implementations in this paper are able to consider different geometries, driver/receiver strengths, and noise bounds for different nets. For simplicity of presentation, we assume that the coplanar interconnect structure is uniform in terms of wire width and space, driver/receiver strength, and we find a SINO solution where the peak noise among all nets satisfies a given noise voltage bound.

3 RLC interconnect modeling

In this section, we discuss first the RLC circuit model for coplanar interconnect structures, and then the noise model for coupled RLC interconnects. Our noise model computes the peak noise voltage for the victim induced by the worst-case test pattern for all aggressors.

3.1 RLC Circuit Model

In our circuit model, each driver is modeled by a driver resistor R_d , and each receiver by a loading capacitance C_L . A wire is modeled in general by multiple RLC segments. An example of our RLC circuit model for three coupled wires (also called a three-net structure) is given in Figure 5, where one RLC segment is used for a wire, and R_i , L_i , and C_{gi} ($i = 1, 2$, and 3) are resistance, self inductance and ground capacitance for the three wires. We consider coupling capacitance only for adjacent wires, and consider coupling inductance for any two wires. For the three-net structure in figure 5, there are coupling capacitance and coupling inductance (Cx_{12} and Cx_{23} , Lm_{12} and Lm_{23}) between adjacent *wire*₁ and *wire*₂, and adjacent *wire*₂ and *wire*₃. In addition, there is coupling inductance Lm_{13} between nonadjacent *wire*₁ and *wire*₃ (but not coupling capacitance between them). We obtain capacitance and inductance from interconnect geometry using the table-based models for capacitance and inductance presented in [22] and [20], respectively. As pointed out in [20, 23], we use the partial inductance model without assuming any current return path in order to achieve an accurate RLC circuit model. In this paper, one RLC-segment is used for each wire in noise computation, but multiple RLC-segments are used for each wire in SPICE simulations.

3.2 RLC Noise Computation

As the coupling inductance is a “long-range” effect (see Figure 1), the peak noise for the victim under the RLC circuit model is affected virtually by all aggressors in a coplanar interconnect structure. Even though the one-segment RLC model is used for each wire, the resulting RLC circuit model (called the *full model*) is still too complicated to be analyzed efficiently for a wide interconnect structure such as a 32-bit bus. We propose to decompose a wide interconnect structure into a number of three-net structures, then obtain the peak noise of the victim by solving these three-net structures. The key is to guarantee that the noise computed via three-net structures be conservative when compared to the noise computed based on the full model. If so, the conservative noise can be used effectively to guide interconnect synthesis. Further, the resulting noise model should be efficient enough for interconnect synthesis, as three-net structures can be solved much more efficiently compared to wide interconnect structures.

In the following, we proceed to present first the solution to the three-net structure, then the solution to a wide interconnect structure. We will also discuss the inductive screening rule to find all aggressors for the given victim, and use SPICE simulations to show that our noise model is conservative. For all experiments in this paper, we assume that all wires are $1.1\mu m$ thick as typical global interconnects in the $0.1\mu m$ process given in [24]. Additionally, the Vdd is $1.05V$, the input rising time is $33ps$, the equivalent driver resistance is 150Ω , and the load capacitance is $60fF$. Note that one RLC-segment is used to model a net (of any length⁴) in our noise model, but multiple RLC-segments are used to model a net, more specifically, one RLC-segment for each $100\mu m$ wire segment for SPICE simulation.

3.2.1 Three-net structure

We model the three-net structure as the RLC circuit shown in Figure 5. We assume that *wire*₁ is the quiet victim with noise voltage

$$V_{out_{victim}} = H_2 \cdot V_{s2} + H_3 \cdot V_{s3} \quad (1)$$

where V_{s2} and V_{s3} are the input to *wire*₂ and *wire*₃, respectively, and H_2 and H_3 are their transfer functions in the s-domain. Both H_2 and H_3 have the following form

$$H_{2 \text{ or } 3} = \frac{a_0 + a_1 \cdot s + a_2 \cdot s^2 + a_3 \cdot s^3 + a_4 \cdot s^4}{b_0 + b_1 \cdot s + b_2 \cdot s^2 + b_3 \cdot s^3 + b_4 \cdot s^4 + b_5 \cdot s^5 + b_6 \cdot s^6} \quad (2)$$

The numerator and denominator coefficients are given in the Appendix A for simplicity of presentation. We use a five-pole model to first solve H_2 and H_3 , and then obtaining solution to Eqn. (1) by using the superposition rule. We always assume the worst-case signal pattern, either *vqa* or *vaa* for the three wires. *vqa* means that victim is quiet, *wire*_{*j*+1} is quiet or is a shield, and *wire*_{*j*+*i*} is a switching aggressor. *vaa* means that victim is quiet, both *wire*_{*j*+1} and *wire*_{*j*+*i*} are aggressors which switch in the same direction.

In Table 1, we compare the peak noise generated by our noise model and SPICE simulation. In this experiment, we assume that all wires are $3000\mu m$ long, $1\mu m$ wide, and are separated by an edge-to-edge spacing of $1\mu m$. As shown in this Table, our noise model is conservative when compared to SPICE simulation.

⁴Due to delay constraint, interconnects are in general broken by buffers into segments no longer than $4000\mu m$.

signal pattern	noise model	SPICE
<i>vqa</i>	0.0953	0.0735
<i>vaa</i>	0.345	0.299

Table 1: Peak noise voltage generated by our model and SPICE. *vqa* means that *wire*₁ is the quiet victim, *wire*₂ is a quiet wire or a shield, and *wire*₃ is a switching aggressor, while *vaa* means that *wire*₂ and *wire*₃ are both aggressors which switch in the same direction in order to generate the peak noise.

3.3 Wide interconnect structure

3.3.1 Basic Algorithm

We use the following algorithm to compute the noise for the victim within a wide interconnect structure. As illustrated in Figure 6, we compute the noise side by side for the victim *wire*_{*j*}. For the upper side, the noise contribution of *wire*_{*j*+1} and *wire*_{*j*+2} can be easily given by the three-net structure containing *wire*_{*j*}, *wire*_{*j*+1}, and *wire*_{*j*+2}. Note that the resulting noise is conservative, as the current generated by *wire*_{*j*+1} and *wire*_{*j*+2} in the full model (with modeling of all nets) may return from nets other than the above three-net structure and therefore introduce less noise for the victim.

Then, for any *wire*_{*j*+i} (*i* > 2), if the wire is sensitive to the victim, we construct a three-net structure containing *wire*_{*j*}, *wire*_{*j*+1} and *wire*_{*j*+i} (see Figure 6). The noise contribution of *wire*_{*j*+i} will be computed by $H_3 \cdot V_{s3}$ as in Eqn. (1). Again, this noise contribution is conservative because the constructed three-net structure ignores current return path via nets other than *wire*_{*j*} and *wire*_{*j*+1}. Further, the coupling capacitance between *wire*_{*j*+i} and *wire*_{*j*+1} is zero for *i* > 2. In addition, as shown in Table 1, we always assume the worst-case signal pattern, either *vqa* or *vaa* for the three wires, depending on *wire*_{*j*+1} being quiet or switching.

Finally, nets in the lower side of the victim can be processed similarly. The absolute noise values from all three-net structures are added to give the peak noise for the victim in the wide interconnect structure. Additionally, an inductive screening rule is presented below to decide which aggressors can be ignored to avoid over-estimation.

3.3.2 Screening of inductive coupling

Considering aggressors only in the current block, as in the K_{eff} model used in [13] may under-estimate the peak noise, while considering all aggressors in a wide interconnect structure may over-estimate induced noise for the victim. The following screening rule is presented very recently in [25]: The potential victim wires of an inductive aggressor (or a group of simultaneous aggressors) are those nearest neighboring wires with their total width equal to or less than twice the width of the aggressor (or the total width of the aggressors).

Stimulated by this aggressor-oriented screening rule, we define the following victim-oriented screening rule for our noise computation and SINO algorithms: Let K_s be the screening constant, $W_a(a_i)$ is the accumulative wire width for aggressors between the victim and aggressor a_i (including a_i), and $W_q(a_i)$ is the accumulative wire width for quiet wires or shields between the victim and aggressor a_i (excluding the quiet victim). If $W_a(a_i) \cdot K_s \leq W_q(a_i)$, aggressor a_i does not contribute to the inductive noise induced in the victim. Note that our peak noise model *always* considers aggressors in the current block, i.e., we only use the screening rule for

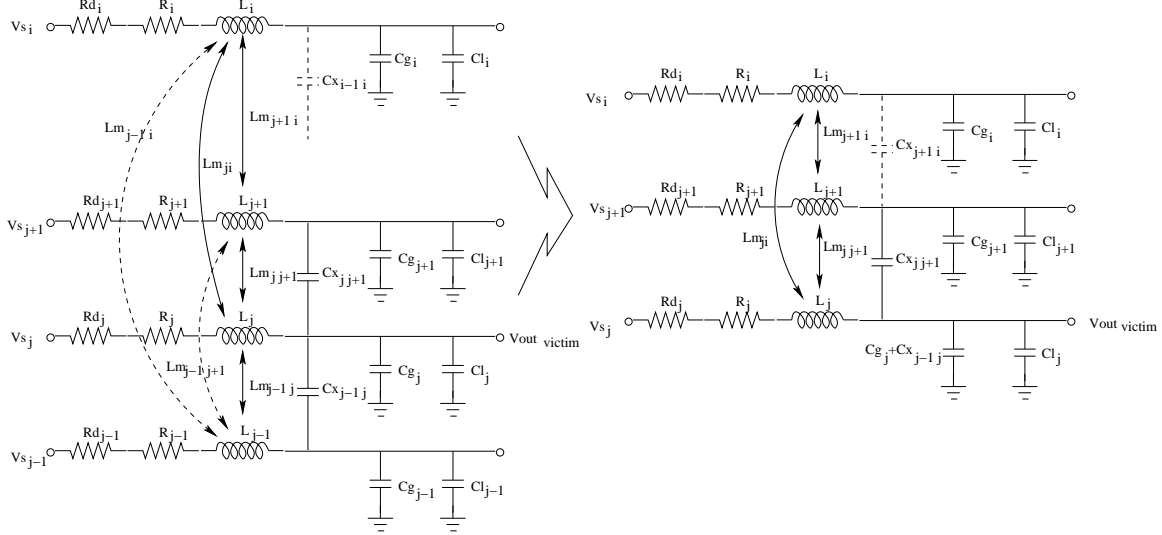


Figure 6: The mapping of a wide interconnect structure with specific victim and aggressors to the three-net model for calculation of the noise generated in the victim due to the aggressor.

aggressors outside the current block.

The new screening rule can be illustrated by the following SINO string:

$$a_1 q a_2 a_3 a_4 q v a_5 q a_6$$

where a_1, \dots, a_6 are aggressors, v is the victim, and q 's are either quiet wires or shields. We assume that all wires have the same width w . When the screening constant $K_s = 0.5$, for aggressor a_4 , $W_a(a_4) = w$, $W_q(a_4) = w$, therefore $0.5W_a(a_4) < W_q(a_4)$ and a_4 does not contribute to the noise induced in the victim; for aggressor a_2 , $W_a(a_2) = 3w$, $W_q(a_2) = w$, therefore $0.5W_a(a_2) > W_q(a_2)$ and a_2 contributes to the noise induced in the victim.

3.3.3 Experimental verification

In Table 2, we compare our peak noise model according the above algorithm to the peak noise generated by SPICE simulation using detailed RLC circuit model where each $100\mu m$ -long wire is modeled by an RLC segment. We consider wide structures with from eight to thirty-two nets. We assume that all nets are $3000\mu m$ long, and their wire width and spacing are randomly generated for each net. The width is between $0.5\mu m$ and $3\mu m$, and the center to center spacing is between 1.5 and 4 microns. Capacitance and inductance values are generated using table-based models according to the specific width and spacing. The signal patterns used by SPICE simulations are given in the table, where v is the quiet victim, q can be either a quiet wire with driver and receiver or a shield directly connected to power supply networks for every $250\mu m$ -long wire segment, and a is a switching aggressor. We assume that all aggressors have the same-direction simultaneous switching for SPICE simulation. As shown in this Table, our model is consistently conservative, with from 7.2% to 51.3% overestimation. We did not apply the screening rule in this experiment. This may contribute to the higher overestimation rate here.

Bus size	Peak Noise(Volt)			Signal Pattern
	Noise Model	SPICE	% Over-estimation	
8	0.336	0.298	12.8%	aaaaqvqa
8	0.23	0.223	7.2%	aaqvqaaa
16	0.543	0.359	51.3%	aaqaaqavvqaaqaaaq
16	0.431	0.310	39.0%	aaqaaqqvqqqaaaq
32	0.725	0.509	42.4%	aaqaaaqaaaaaqaavvqaaqaaaaaaqaaq
32	0.291	0.193	50.8%	qqaaqqaaqqvqaaqaqqqqaaqqaaqqaaqq

Table 2: Comparison of our noise model and SPICE-computed peak noise for wide interconnect structures.

4 Properties and Algorithms

4.1 Property for SINO/NB Problem

Theorem 1 *The optimal SINO/NB- v problem is NP-hard.*

Sketch of proof: The SINO/NB- v problem and the SINO/NB- K_{eff} problem are essentially the same, with the only difference between them being computational costs of noise model (a polynomial time operation). Since the SINO/NB- K_{eff} problem is NP-hard (shown in [13]), and adding a polynomial-time operation does not affect NP-hardness, the SINO/NB- v problem is also NP-hard.

Because SINO/NB is a computationally intractable problem, we focus on heuristic methods to obtain high-quality solutions with reasonable computation time.

4.2 Algorithms for Solving SINO/NB Problems

We examine and extend three of the approximate algorithms originally developed in [13] for solving the SINO/NB- k problem: greedy-based shield insertion (SI) algorithm, net ordering for minimizing Cx noise followed by SI algorithm (NO+SI algorithm), and simulated-annealing based SINO algorithm (SINO algorithm). However, these SINO/NB- v algorithms are more complicated than those SINO/NB- k algorithms using the K_{eff} model. First, our peak noise model may consider aggressors beyond the current block whereas the K_{eff} model considers aggressors only within the current block. Secondly, the victim may be placed directly adjacent to an aggressor in the SINO/NB- v formulation but not in the SINO/NB- k formulation.

The SI and NO+SI algorithms can be described easily and intuitively. SINO/SA is slightly more complicated, hence we assume that the reader is familiar with SA (for a discussion of SA in other VLSI design contexts, see [26], [27]).

4.2.1 Greedy Based Shield Insertion Algorithm

The essence of the greedy-based shield insertion (SI) algorithm is the following: Run through the given placement P , at first ignoring the inductance screening rule presented in 3.3.2. At each location in the placement, calculate the maximum value of N_i that would exist in the current block if we allowed net s_i to become a member. If N_i is greater than N_{thresh} , then create a new block.

To consider inductance screening, we then run a second pass over the interconnect structure, finding the maximal N_i . If the noise for this net $s_i(N_i)$ is greater than N_{thresh} , we insert a shield directly next to s_i in the placement in the track that reduces N_i most. We continue this process of finding the maximal N_i and inserting shields (within the successively improving structure) until all $N_i \leq N_{thresh}$.

The solution given by the SI algorithm depends on the initial placement. If sufficient capacitive coupling exists, obviously, the number of shield wires can be reduced by first running existing net ordering algorithms to re-order nets so that no sensitive nets are adjacent to each other and subsequently invoking the SI algorithm. This leads to the NO+SI algorithm.

4.2.2 Simulated Annealing Based SINO Algorithm

In Figure 7, we present the simulated-annealing based SINO (SINO/SA) algorithm. We give the details of our SINO/SA algorithm in the following subsections:

Cost Function *Compute_Cost(P)* computes the cost for a placement P . The cost is the weighted sum of the following components: (i)*Area*: total number of g-wires present in P ; (ii)*Noise*: total number of $N_i > N_{thresh}$ violations in P ; and (iii)*Noise Violation Figure*. It is computed for a placement as shown in Figure 8. The purpose of the Noise Violation Figure is to penalize a placement for the magnitude of N_{thresh} violations. Its usage (as opposed to simply forbidding placements P' that have N_{thresh} violations) allows the algorithm to potentially trade noise violations for smaller overall placement size depending on the result desired, and can be useful in different SINO formulations. It is also worthwhile to note that this definition of Noise Violation Figure differs slightly from the Inductance Violation Figure given in [13]. The change was made due to scaling issues—the values for N_{thresh} are much lower than the values for K_{thresh} used in [13], and using the Inductance Violation Figure as originally proposed causes anomalous behavior in the SA algorithm. The weighting factor for each cost component can be tuned for different design objectives. In this paper our stated goal is to minimize placement size without violating N_{thresh} noise constraints, hence weighting factors were chosen to help us achieve those goals with maximal efficiency.

Random Moves *Random_Move(P, P')* performs one of the following changes to placement P to generate a new placement P' : (i) Combine two random blocks in P , (ii) Swap two random s-wires in P , (iii) Move a single random s-wire to a new and random location, (iv) Insert a g-wire at a random location in P . It is worthwhile to note that combining two random blocks in a placement P is also equivalent to removing a g-wire if the two blocks are adjacent. Moves which create two adjacent g-wires in a placement are categorically rejected and a new move is tried.

Temperature Adjustment and Stopping Criterion The method of temperature adjustment is shown in Figure 7. We use a simple multiplicative constant of the current temperature. At each temperature step, the variance of the current placement cost from its previous value is taken and averaged over several random moves to determine the stability of the system at each temperature. When the variance is less than a set threshold, we move to the next temperature step. The starting temperature, freezing point, temperature adjustment, and variance threshold factors were all determined experimentally.

```

Simulated Annealing Algorithm: Given a placement  $P$ :
Repeat
  Temp = Initial_Temperature;
  Repeat
     $Random\_Move(P, P')$ ;
     $Candidate\_Cost = Compute\_Cost(P')$ ;
     $ds = Candidate\_Cost - Compute\_Cost(P)$ ;
    if ( $ds < 0$ )
       $P = P'$ ;
    else
       $r = RANDOM(0, 1)$ ;
      if ( $r < exp(-ds/Temp)$ )
         $P = P'$ ;
  Until equilibrium at Temp is reached;
   $Temp = Temp * Temperature\_Adjustment$ ;
  /*( $0 < Temperature\_Adjustment < 1$ )*
Until Temp == Freezing_Point;

```

Figure 7: Simulated Annealing SINO Algorithm (SINO/SA)

```

 $Compute\_Noise\_Screening(s_i)$  {
  Compute the accumulated noise induced in  $s_i$  by all aggressors
  decided by the inductive coupling screening rule in  $P$  }
 $Total\_Violation\_Figure = 0$ ;
for each  $s_i$  in placement  $P$ 
  if ( $N_i = Compute\_Noise\_Screening(s_i) > N_{thresh}$ )
     $Total\_Violation\_Figure += ((1 + N_i - N_{thresh})^3) - 1$ 
end

```

Figure 8: Computation of the Noise_Violation_Figure

4.2.3 Algorithm Speedup

Because the complexity of the explicit noise computation model used in this work is substantially greater than the K_{eff} model presented in [13], we use a table to speed the explicit noise computation. Because of symmetry among the wires in a uniform coplanar interconnect structure, a table of three dimensional in block_size and the aggressor and victim locations within the block is sufficient. When we wish to determine the noise voltage in a given s-wire, we simply look up the geometry (block size) in the table. If no entry for that block size exists in the table, we compute all possible combinations for the block size (because of the way the current SINO algorithm works, it is very likely that most of these entries will be used in short order, hence there is little wasted work) and return the noise voltage for the wire as described previously. If the table entry has already been computed, we simply lookup the value in the table. We do not pre-compute the table, but the table is shared across multiple interconnect runs, and hence the cost of building it can be amortized. In general, when the interconnect structure is not uniform, more dimensions may be needed by the table. For an aggressor outside the current block, the block size used in the table is the size of the super-block that comprises of blocks

from the one containing the victim to the one containing the aggressor.

We can also improve the performance of our SINO algorithms by exploiting the highly efficient K_{eff} model developed previously. This is best described as it relates to the SA algorithm, but this technique is not only useful for this algorithm. We run simulated annealing in two phases. We start the process by setting an appropriate K_{thresh} and running SINO/SA-Keff for some period of time (in our current implementation, it is until a preset temperature, determined experimentally). We then switch to the SINO/SA-Noise algorithm so that we can approximate noise voltage directly to run the low temperature annealing passes. This method improves runtime because we leverage the highly efficient K_{eff} model at high temperatures (where accuracy is not paramount), and we quickly proceed to lower temperatures where the more accurate (and slower) explicit noise model can be used. This method can also save running time because of fewer lookup table row computations by eliminating large row computations that are present early in the annealing process (if we start from an unshielded initial placement).

5 Experimental Results and Discussions

In this section, we first present and analyze the experiment results, then discuss the implications to future multi-GHz designs.

5.1 Tool implementation and experiment setting

We have implemented an integrated toolset in the C++ programming language. The toolset includes the table-based models for capacitance and inductance proposed in [22] and [20], all shield insertion and net ordering algorithms proposed in [13] and in this paper, and SPICE netlist generation. The user may specify the geometry of the coupled nets to be optimized, the driver resistance and loading capacitance for each net, the noise bound for each net, and the algorithm to apply. The SPICE netlist can be automatically generated and be used to verify the interconnect optimization result. The toolset can be invoked interactively or via a command script.

We have tested our algorithms and implementations using a large number of examples. In this section, we use a coplanar interconnect structure containing 32 s-wires as the testing example for different combinations of noise bound, interconnect geometry, and sensitivity rate. We consider the following noise constraints: 0.15V, 0.20V, 0.25V. These constraints are respectively 14%, 19% and 23% of the power supply voltage $V_{dd} = 1.05V$ in the $0.1\mu m$ process given in [24]. Because the noise is measured at the input of the receiver and receiver may serve as a noise filter, we allow a noise bound of $V_{dd} \cdot 23\%$ in this experiment. Additionally, we assume that the input rising time is $33ps$ (10% of the clock period in a 3GHz clock for the $0.1\mu m$ process), the equivalent driver resistance is 150Ω , and the load capacitance is $60fF$. We set our screening constant $K_s = 0.33$.

For all experiments in this section, we assume that all wires are $1.1\mu m$ thick, $1\mu m$ wide, and $2000\mu m$ long. We consider two edge-to-edge spacings of $0.8\mu m$ and $1.6\mu m$, and two sensitivity rates of 30% and 60%. When the sensitivity rate is 30%, each net is sensitive to 30% of all nets, and these sensitive nets are picked randomly for the given s-wire.

We generate twenty different sensitivity matrices and initial placements for each design combination of noise bound, geometry and sensitivity rate. We consider the following algorithms: greedy-based shield insertion (SI), net ordering followed by SI (NO+SI), and simulated-annealing based SINO (SINO), and present experimental results in Tables 3 to 5. To make our analysis reliable and the comparisons “fair” among the different algorithms,

noise bound	spacing = $0.8\mu m$			spacing = $1.6\mu m$		
	SI	NO+SI	SINO	SI	NO+SI	SINO
length = $2000\mu m$, sensitivity rate = 30%						
0.15V(14%Vdd)	15.05(201%)	6.85(37%)	5.00(0%)	7.50(63%)	5.95(27%)	4.70(0%)
0.20V(19%Vdd)	14.75(201%)	5.95(21%)	4.90(0%)	6.80(55%)	5.15(17%)	4.40(0%)
0.25V(23%Vdd)	14.15(249%)	5.45(35%)	4.05(0%)	6.15(50%)	5.15(26%)	4.10(0%)
length = $2000\mu m$, sensitivity rate = 60%						
0.15V(14%Vdd)	25.70(222%)	10.50(32%)	7.95(0%)	14.35(75%)	10.45(27%)	8.20(0%)
0.20V(19%Vdd)	24.25(219%)	10.05(32%)	7.60(0%)	13.90(79%)	9.75(42%)	7.75(0%)
0.25V(23%Vdd)	23.80(230%)	9.55(33%)	7.20(0%)	12.06(70%)	10.30(39%)	7.40(0%)

Table 3: Number of shields (including two edge shields) inserted by algorithms SI, NO+SI and SINO.

SPICE-computed maximum/average peak noise (V)						
noise bound	spacing = $0.8\mu m$			spacing = $1.6\mu m$		
	SI	NO+SI	SINO	SI	NO+SI	SINO
length = $2000\mu m$, sensitivity rate = 30%						
0.15V	0.0458/0.0332	0.0962/0.0625	0.0939/0.0737	0.1193/0.0822	0.1236/0.0876	0.1357/0.0942
0.20V	0.0738/0.0400	0.1120/0.0774	0.0913/0.0726	0.1342/0.1001	0.1388/0.0971	0.1407/0.0984
0.25V	0.0617/0.0376	0.1040/0.0825	0.1230/0.0985	0.1278/0.0810	0.1509/0.1087	0.1482/0.1062
length = $2000\mu m$, sensitivity rate = 60%						
0.15V	0.0495/0.0341	0.1099/0.0956	0.1091/0.0776	0.1338/0.0960	0.1107/0.0892	0.1357/0.1138
0.20V	0.0436/0.0345	0.1225/0.1082	0.1141/0.0890	0.1619/0.1251	0.1489/0.0962	0.1620/0.1291
0.25V	0.0573/0.0372	0.1299/0.1139	0.1287/0.0916	0.1948/0.1649	0.1757/0.1008	0.1683/0.1434

Table 4: SPICE-computed maximum and average peak noise for interconnect synthesis solutions given by different algorithms. In each cell of the table, the first value is the maximum peak noise, and the second value is the average peak noise.

we run each algorithm on these twenty cases for each design combination, and report the statistic results based on these twenty runs for each design combination. Finally, it is worthwhile to point out that we did not tune the screening constant K_s and our SA-based SINO algorithm for different examples.

5.2 Analysis of Experimental Results

5.2.1 Algorithm Validation

The peak noise value is the maximum noise among all wires in a SINO solution considering the worst-case signal patterns. To compute the peak noise, we generate a detailed RLC circuit model for each SINO solution, with an RLC-segment for each $100\mu m$ -long wire segment and a mutual inductance between any two such wire segments. The partial inductance model is used to achieve high accuracy. The resulting RLC circuits are simulated by SPICE running on a network of workstations. We report the maximum and average peak noise

for twenty randomly sampled nets for each design combination (random sampling is necessary to save SPICE runtime even though we use a network of workstations).

Table 4 presents maximum and average peak noise values obtained by our accurate verification. One can easily see that the peak noise for all 720 test cases presented in this paper satisfies the given noise bound, and there is a smooth tradeoff between the resulting average number of shields and average peak noise. This validates our peak noise model, the SINO/NB- v problem formulation, and the toolset implementation.

5.2.2 Algorithm Quality

As all solutions meet the given noise bounds, the algorithm quality is measured by the number of shields needed for the given noise bound. Table 3 presents numbers of shields used by different algorithms. As we anticipate, SI is always worse than NO+SI and SINO, and SINO is always best. The spacing greatly affects the evaluation of algorithm quality. For the smaller spacing ($0.8\mu m$) which is more likely to be used in real designs, SINO uses up to 37% and 249% fewer shields when compared to NO+SI and SI, respectively. For the larger spacing ($1.6\mu m$), SINO uses up to 27% and 142% fewer shields when compared to NO+SI and SI, respectively. The SA-based SINO algorithm is clearly the best choice for a real tool implementation in terms of solution quality.

5.2.3 Runtime

Because the SA-based algorithm gives the best solutions, we present the runtime only for the SA-based algorithm in Table 5. We collected the runtime for cases with wire spacing of $0.8\mu m$, and did not observe practically noticeable differences for cases with other spacings. We have implemented two versions of the SA-based algorithm: one is guided purely by the noise voltage model, and the other is guided first by the K_{eff} model and then by the noise voltage model (see Section 4.2.3). The two are denoted by SA-Voltage and SA-Hybrid in the Table, respectively. Further, as solutions to three-net structures are pre-computed and stored in tables for reuse, we present the runtime for one run, each time computing the whole lookup table as needed, as well as the average runtime per run for 10 runs in a row, computing the whole lookup table only once. The runtime was collected on an Intel 933MHz PIII workstation. As shown in the Table, the runtime per run is about ten seconds, and can be further reduced when multiple runs are invoked in a row such that costs to build noise tables are amortized. The SA-Hybrid version uses less runtime in all cases, ranging from approximately 25% to 5% less. Additionally, the SA-Voltage and SA-Hybrid achieve SINO solutions that have statistically-equivalent quality verified in experiments not presented here.

5.3 Implications to Multi-GHz designs

According to the best solution given in Table 3 for 60% sensitivity rate and $0.8\mu m$ spacing, i.e., when 60% of signals are switching in the same time window for a $2000\mu m$ -long 32-bit bus with $1\mu m$ wire width, on average 7.5 shields are needed to bring the SPICE-computed noise down to about 10% of Vdd. This is approximately 23% area overhead for shielding purposes. Intuitively, the area overhead is going to increase for higher clock frequencies and therefore quicker signal rising times. We speculate that shield insertion would play an increasingly important role for future multi-GHz designs.

Simultaneous shield insertion and net ordering is a key to obtaining an area-efficient solution. Greedy-based increasing of spacing and shield insertion is a common method for noise minimization. For our 32-bit

	SA-Voltage			SA-Hybrid		
noise bound (V)	0.15	0.20	0.25	0.15	0.20	0.25
sensitivity rate	runtime per run					
30%	8.6	9.5	11.8	7.8	6.9	9.1
60%	13.8	14.8	14.7	13.1	13.6	11.8
sensitivity rate	runtime per run (average over 10 runs)					
30%	3.9	4.4	5.6	3.1	3.4	4.5
60%	3.7	4.7	5.3	3.2	4.0	4.0

Table 5: Runtimes in seconds for SA-based algorithms.

bus example with 0.15V noise bound and 60% sensitivity rate, the SINO solution with $1\mu m$ wire width, $0.8\mu m$ spacing and (on average) 7.5 shields has a total width of $(32 + 7.5) \cdot (1 + 0.8) = 71.1\mu m$. At the same time, the greedy-based SI solution with $1\mu m$ wire width, $1.8\mu m$ spacing and (on average) 8.45 shields has a total width of $(32 + 8.45) \cdot (1 + 1.6) = 105\mu m$. The SINO solution saves about 34% in total area. Further, as shown in Table 3, the area overhead of shielding increases when the sensitivity rate (the chance of simultaneous switching signals) increases. To minimize the overhead due to shielding, the chance of simultaneous switching signals should be minimized for future multi-GHz design at both RTL and physical design levels. Bringing down the number of simultaneous switching signals also contributes to minimizing noise for power delivery networks.

6 Conclusions and Future Work

For a number of coupled RLC nets, we have formulated the min-area simultaneous shield insertion and net ordering (in short, SINO/NB- v) problem to satisfy the given bound on noise voltage. We have developed an efficient model to compute the peak noise among coupled RLC nets, and have applied the noise model to a simulated-annealing (SA) based algorithm for the SINO/NB- v problem. Accurate verification show that the SA-based algorithm always achieves the SINO solution satisfying the given noise bound, and uses up to 249% and 37% fewer shields when compared to a greedy based shield insertion algorithm and a separated shield insertion and net ordering algorithm. Further, the SA-based algorithm is efficient to optimize a 32-bit bus in approximately 10 seconds.

The simplest interconnect structure, parallel bus structure is assumed in this work. We are working on the SINO/NB- v problem for the multi-layer routing model with regular P/G networks, and intend to further incorporate the SINO/NB- v formulation into the global routing problem.

Appendix A: Solution the three-net structure

The numerator and denominator coefficients in Eqn.(2) are functions of resistance R_d for the driver, loading capacitance Cl for the receiver, and resistance R , self inductance L , coupling inductance Lm , ground capacitance Cg and coupling capacitance Cx for the wire. Transfer functions H_2 and H_3 are only different in these coefficients. Fortunately the denominators of both H_2 and H_3 are the same, which leads to an even simpler implementation. The followings are the formula for the coefficients for the three-net structure with uniform

wire width. Note that our implementation is able to consider non-uniform cases. For simplicity of presentation, we below lump the C_l and C_g into C , and lump Rd and R into R' .

$$\begin{aligned}
b_0 &= 1. \\
b_1 &= R'C_1 + 4R'Cx + R'C_2 + R'C_3. \\
b_2 &= -4LmCx + C_2R'^2C_3 + 4LCx + LC_2 + 2C_2R'^2Cx + 3Cx^2R'^2 + 3CxR'^2C_3 + LC_1 + R'^2C_1C_3 \\
&\quad + 3R'^2C_1Cx + LC_3 + R'^2C_1C_2. \\
b_3 &= CxR'^3C_2C_3 + 2R'C_1LC_3 + 2C_2LR'C_3 - 4LmCx^2R' + R'^3C_1C_2C_3 + Cx^2R'^3C_2 - 2CxLmR'C_3 \\
&\quad + 6CxLR'C_3 + 6R'C_1LCx + Cx^2R'^3C_3 + 4C_2R'LCx - 2Cx^2Lm_{13}R' + 2R'C_1C_2L - 2R'C_1LmCx \\
&\quad + 2R'^3C_1CxC_3 + Cx^2R'^3C_1 + 6Cx^2R'L + R'^3C_1C_2Cx. \\
b_4 &= 2CxLmC_3Lm_{13} - 2CxLmLC_3 - 2CxLm^2C_3 - 2Lm^2C_2Cx - 4LmCx^2L + 4Lm_{13}Cx^2Lm \\
&\quad - 2Cx^2Lm_{13}L + 3Cx^2L^2 + 3CxL^2C_3 - 2Lm^2C_1Cx - Lm_{13}^2CxC_1 + 2C_2L^2Cx + 3L^2C_1Cx \\
&\quad + 6R'^2C_1CxLC_3 + 3R'^2C_1C_2LCx + 3CxR'^2C_2LC_3 + 3Cx^2R'^2C_2L + 3Cx^2R'^2C_1L + 3Cx^2R'^2LC_3 \\
&\quad - 2LC_1LmCx + 2LmCxLm_{13}C_1 - Lm_{13}^2Cx^2 - CxLm_{13}^2C_3 - Lm^2C_2C_3 - Lm^2C_1C_2 \\
&\quad - Lm_{13}^2C_3C_1 + C_2L^2C_3 + L^2C_1C_3 + L^2C_1C_2 + 3R'^2C_1C_2LC_3. \\
b_5 &= -2CxLm^2R'C_2C_3 - 4Lm^2R'C_1Cx C_3 + 3R'C_1C_2L^2C_3 - 2Lm^2R'C_1C_2Cx + 6CxR'C_1L^2C_3 \\
&\quad - Cx^2Lm_{13}^2R'C_2 + 3Cx^2R'C_1L^2 - 2Cx^2Lm^2R'C_1 + 3Cx^2R'C_2L^2 + 3R'C_1C_2L^2Cx \\
&\quad - Cx^2Lm_{13}^2R'C_1 - Lm_{13}^2CxR'C_2C_1 + 3CxR'C_2L^2C_3 - 2CxLm_{13}^2C_3R'C_1 - 2Cx^2Lm^2R'C_2 \\
&\quad - 2Cx^2Lm^2R'C_3 - CxLm_{13}^2C_3R'C_2 + 3Cx^2R'L^2C_3 - Lm_{13}^2C_3R'C_2C_1 - 2Lm^2R'C_1C_2C_3 \\
&\quad - Cx^2Lm_{13}^2C_3R'. \\
b_6 &= Cx^2L^3C_1 + Cx^2L^3C_3 + Cx^2L^3C_2 + 2CxL^3C_1C_3 - 2Cx^2Lm^2LC_2 - 2Cx^2Lm^2LC_3 \\
&\quad + CxL^3C_2C_3 - Cx^2Lm_{13}^2C_3L - Cx^2Lm_{13}^2LC_2 - Cx^2Lm_{13}^2LC_1 + L^3C_1C_2Cx - 2Cx^2Lm^2LC_1 \\
&\quad - 4CxLm^2LC_1C_3 + 4Lm^2CxLm_{13}C_1C_3 + 2Lm^2C_2Lm_{13}C_1Cx + 2CxLm^2C_2Lm_{13}C_3 \\
&\quad - 2CxLm^2LC_2C_3 - CxLm_{13}^2C_3LC_2 - 2CxLm_{13}^2C_3LC_1 - 2Lm^2LC_1C_2Cx - Lm_{13}^2CxLC_2C_1 \\
&\quad + 2Cx^2Lm^2Lm_{13}C_3 + 2Cx^2Lm^2C_2Lm_{13} + 2Cx^2Lm^2Lm_{13}C_1 + 2Lm^2C_2Lm_{13}C_1C_3 \\
&\quad - 2Lm^2LC_1C_2C_3 + L^3C_1C_2C_3 - Lm_{13}^2C_3LC_2C_1.
\end{aligned}$$

As it is mentioned before, the denominator coefficients, $b_0\dots b_7$, are the same for both H_2 and H_3 . However, the coefficients for the numerators are different. The coefficients for the numerator of H_2 are listed below.

$$\begin{aligned}
a_0 &= 0. \\
a_1 &= R'Cx. \\
a_2 &= LCx - 2LmCx + CxR'^2C_3 - LmC_2 + CxLm_{13} + Cx^2R'^2. \\
a_3 &= -2LmCx^2R' + 2CxLR'C_3 - LmR'CxC_2 - LmR'C_3C_2 + 2Cx^2R'L - 2CxLmR'C_3. \\
a_4 &= -CxLm_{13}^2C_3 - 2LmCx^2L + LmC_2CxLm_{13} - Lm_{13}^2Cx^2 + 2CxLmC_3Lm_{13} + Cx^2L^2 \\
&\quad - LmLCxC_2 - 2CxLmLC_3 + CxL^2C_3 - LmLC_3C_2 + 2Lm_{13}Cx^2Lm + LmC_2C_3Lm_{13}.
\end{aligned}$$

And the numerator coefficients for H_3 are:

$$a_0 = 0.$$

$$a_1 = 0.$$

$$a_2 = LmCx - CxLm_{13} - C_3Lm_{13} + Cx^2R'^2.$$

$$a_3 = -Cx^2Lm_{13}R' - CxLmR'C_3 - CxLm_{13}C_2R' - C_3Lm_{13}C_2R' \\ - 2C_3Lm_{13}CxR' + 2Cx^2R'L - LmCx^2R'.$$

$$a_4 = CxLmC_3Lm_{13} + 2CxLm^2C_3 + Lm^2C_2Cx - CxLmLC_3 + Cx^2L^2 - Cx^2Lm_{13}L - 2C_3Lm_{13}CxL \\ - CxLm_{13}C_2L - LmCx^2L + Lm_{13}Cx^2Lm - C_3Lm_{13}C_2L + Lm^2C_2C_3.$$

Despite the differences in coefficients, we can solve the inverse Laplace transform of these two functions using the same routine.

References

- [1] *EDA Roadmap Taskforce Report on Design of Microprocessors*. EDA Industry Council, 1999.
- [2] J. Cong, "Challenges and opportunities for design innovations in nanometer technologies," *SRC Design Sciences Concept Paper*, 1997.
- [3] J. Cong, L. He, C.-K. Koh, and P. H. Madden, "Performance optimization of VLSI interconnect layout," *Integration, the VLSI Journal*, vol. 21, pp. 1–94, 1996.
- [4] M. Kamon, M. Tsuk, and J. White, "Fasthenry: a multipole-accelerated 3d inductance extraction program," *IEEE Trans. on MIT*, 1994.
- [5] H. Johnson and M. Graham, *High-Speed Digital Design – A Handbook of Black Magic*. Prentice Hall, 1993.
- [6] J. Lillis, C. Cheng, S. Lin, and N. Chang, *High-performance interconnect analysis and synthesis*. John Wiley, 1999.
- [7] D. Zhou, F. P. Preparata, and S. M. Kang, "Interconnection delay in very high-speed VLSI," *IEEE Trans. on CAS*, July 1991.
- [8] A. Deutsch and et al., "When are transmission-line effects important for on-chip wires," *IEEE Trans. on MIT*, 1997.
- [9] Y. I. Ismail, E. G. Friedman, and J. L. Neves, "Figures of merit to characterize the importance of on-chip inductance," in *Proc. Design Automation Conf*, pp. 560–565, 1998.
- [10] J. Cong and C.-K. Koh, "Interconnect layout optimization under higher-order RLC model," in *Proc. Int. Conf. on Computer Aided Design*, 1997.
- [11] T. Xue, E. S. Kuh, and Q. Yu, "A sensitivity-based wiresizing approach to interconnect optimization of lossy transmission line topologies," in *Proc. IEEE Multi-Chip Module Conf.*, pp. 117–121, 1996.
- [12] Y. I. Ismail and E. G. Friedman, "Effects of inductance on the propagation delay and repeater insertion in VLSI circuits," in *Proc. Design Automation Conf*, pp. 721–724, 1999.
- [13] L. He and K. M. Lepak, "Simultaneous shielding insertion and net ordering for capacitive and inductive coupling minimization," in *Proc. Int. Symp. on Physical Design*, 2000.
- [14] T. Gao and C. L. Liu, "Minimum crosstalk channel routing," in *Proc. Int. Conf. on Computer Aided Design*, pp. 692–696, Nov. 1993.
- [15] T. Gao and C. L. Liu, "Minimum crosstalk switchbox routing," in *Proc. Int. Conf. on Computer Aided Design*, pp. 610–615, Nov. 1994.

- [16] A. Vittal, L. Chen, M. Marek-Sadowska, K. Wang, and S. Yang, "Crosstalk reduction for VLSI," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, March 1997.
- [17] T. Xue, E. S. Kuh, and D. Wang, "Post global routing crosstalk risk estimation and reduction," in *Proc. Int. Conf. on Computer Aided Design*, pp. 302–309, Nov. 1996.
- [18] C.-C. Chang and J. Cong, "Pseudo pin assignment with crosstalk noise control," in *Proc. Int. Symp. on Physical Design*, April 2000.
- [19] R. Kay and R. A. Rutenbar, "Wire packing: A strong formulation of crosstalk-aware chip-level track/layer assignment with an efficient integer programming solution," in *Proc. Int. Symp. on Physical Design*, April 2000.
- [20] L. He, N. Chang, S. Lin, and O. S. Nakagawa, "An efficient inductance modeling for on-chip interconnects," in *Proc. IEEE Custom Integrated Circuits Conference*, pp. 457–460, May 1999.
- [21] S. Khatri, A. M. Mehrotra, R. K. Brayton, and A. Sangiovanno-Vincentelli, "A novel vlsi layout fabric for deep sub-micron applications," in *Proc. Design Automation Conf*, 1999.
- [22] J. Cong, L. He, A. B. Kahng, D. Noice, N. Shirali, and S. H.-C. Yen, "Analysis and justification of a simple, practical 2 1/2-d capacitance extraction methodology," in *Proc. Design Automation Conf*, pp. 627–632, 1997.
- [23] K. Gala, V. Zolotov, R. Panda, B. Y. and Junfeng Wang, and D. Blaauw, "On-chip inductance modeling and analysis," in *Proc. Design Automation Conf*, pp. 63–68, 2000.
- [24] Semiconductor Industry Association, *International Technology Roadmap for Semiconductors*, 2000.
- [25] S. Lin, N. Chang, and O. S. Nakagawa, "Quick on-chip self- and mutual-inductance screen," in *International Symposium on Quality of Electronic Design*, March 2000.
- [26] C. Sechen, "An improved simulated annealing algorithm for row-based placement," in *Proc. Int. Conf. on Computer Aided Design*, pp. 478–481, 1997.
- [27] N. Sherwani, *Algorithms for VLSI Physical Design Automation*. Kluwer Academic Publishers, 1999.