

# Optimal Gate Sizing for Coupling-Noise Reduction

Debjit Sinha, Hai Zhou  
Electrical and Computer Engineering  
Northwestern University

{debjit,haizhou}@ece.northwestern.edu

Chris C N Chu  
Electrical and Computer Engineering  
Iowa State University

cnchu@iastate.edu

## ABSTRACT

Coupling-noise reduction has emerged as a critical design problem with VLSI feature sizes shrinking rapidly and with the use of more aggressive and less noise-immune circuits. Since coupling-noise on a net depends on driving gate-sizes of the net itself and all nets coupled to it, gate-sizing emerges as an effective approach to coupling-noise reduction. It is an attractive approach since re-routing is not required. In this paper, we propose an iterative gate-sizing algorithm to determine *optimal gate-sizes* for coupling-noise reduction. We consider gate-sizing as a fixpoint computation on a complete lattice and the beauty of the iterative gate-sizing algorithm lies in its ability to guarantee the optimal solution, provided it exists. The effectiveness of the algorithm is validated experimentally by simulations on multiple large circuits.

## Categories and Subject Descriptors

J.6 [Computer Aided Engineering]: Computer-aided design (CAD); I.6.5 [Simulation and Modeling]: Model Development; G.2.m [Discrete Mathematics]: Miscellaneous

## General Terms

Algorithms

## Keywords

Coupling-Noise, Gate-Sizing, Lattice Theory, Fixpoint

## 1. INTRODUCTION

With the progress of deep sub-micron technologies, shrinking geometries have led to a reduction in self-capacitance of wires. Meanwhile coupling capacitances have increased as wires have a larger aspect ratio and are brought closer together. For present day processes, the coupling capacitance can be as high as the sum of the area capacitance and the fringing capacitance, and trends indicate that the role of

coupling capacitance will be even more dominant in the future as feature-sizes shrink [1]. This makes coupling-noise a major problem in IC design.

The coupling-noise induced on a net is dependent on the size of its driving-gate and the driving-gate size of each net coupled to it. The noise can be reduced if the size of its driving-gate is increased, or if driving-gate sizes of the coupling nets are decreased. However, when the driving-gate of a net is sized up, it may increase the noise it induces on other nets as an aggressor. On the other hand, when the driving gate-size of an aggressor is reduced, coupling-noise on itself may increase. Since coupling is symmetric on the coupled nets, it is artificial to classify them into aggressors or victims. A net could be an aggressor and a victim at the same time. Though it is plausible to classify a net either as an aggressor or a victim based on the strength of the noise on itself and other coupled nets, with changing driving-gate size, the role of a net may change. Therefore, the classification of nets into an aggressor group or a victim group and the use of attacking edges in the noise graph in the approach by Becer et al. [2] impose limitations.

Crosstalk aware static timing analysis has been used by Xiao et al. [9] in a gate-sizing method for elimination of crosstalk induced timing violation. Another gate-sizing algorithm for crosstalk-noise optimization has been proposed by Hashimoto et al. [7], but is limited by the fact that it only incorporates sizing down the aggressor gates. The algorithm proposed by Becer et al. [2] does not guarantee an optimal solution either. We propose an iterative gate-sizing algorithm to reduce coupling-noise, which is guaranteed to converge to the optimal solution, if it exists. We have considered gate sizing as a fixpoint computation [10] and the beauty of the approach lies in its ability to guarantee the optimal solution, provided it exists. In our approach, we formulate the optimal gate-sizing problem for coupling-noise reduction, and translate the problem into a fixpoint computation problem. A Gate-Sizing transformation is formulated, the solutions to which are mapped onto a complete lattice. We use an iterative algorithm to obtain the optimal solution. The effectiveness of the algorithm is validated by simulations on multiple large circuits.

## 2. PROBLEM FORMULATION

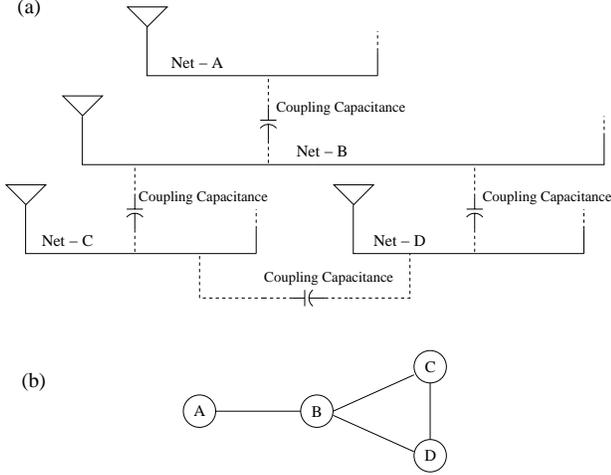
### 2.1 Coupling Graph

We use a *coupling-graph* to model a circuit and to represent coupling information. The coupling-graph is an undirected graph, and may contain cycles. Nodes in the graph represent nets of the circuit, and edges between any two nodes represent coupling between the nets corresponding to the nodes in the coupling-graph. An edge in the coupling-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ISPD'04, April 18–21, 2004, Phoenix, Arizona, USA.  
Copyright 2004 ACM 1-58113-817-2/04/0004 ...\$5.00.

graph is called a coupling-edge and indicates significant coupling capacitance between the two nets it connects. In the extreme case an edge may be introduced between two nets if they have any coupling.



**Figure 1: (a) A sample circuit with 4 nets and coupling capacitances between them. (b) The corresponding Coupling-Graph.**

Figure 1(a) shows a circuit with nets having coupling capacitances between them. Figure 1(b) shows the corresponding coupling-graph where the nodes represent the nets, and coupling-edges are introduced between coupled nets. We use this coupling-graph model in our approach to determine optimal gate-sizes for coupling-noise minimization.

## 2.2 Problem definition

For all mathematical relations and expressions in the paper, we use the syntax of single notation quantification, similar to the one used by *Gries and Schneider* [6]. The general form of a quantification over a binary operator  $\star$  is exemplified by

$$(\star x : R : P)$$

where variable  $x$  is called the *bound variable* or *dummy* of the quantification,  $R$  is a boolean expression called the *range* of the quantification such that values assumed by  $x$  satisfy  $R$ , and  $P$  is an expression which is called the *body* of the quantification. The above expression denotes the application of the operator  $\star$  to the values of  $P$  for all  $x$  for which range  $R$  is *true*. For example, we would represent  $\sum_{i=1}^n x_i$  as  $(+i : 1 \leq i \leq n : x_i)$ .

Given the size of the gates in a circuit, the noise induced on each net can be calculated using a noise-model. It is not necessary that the noise-model be an analytical one. Based on the coupling-graph model as introduced above, the noise  $N(i)$  on a given net  $i$  can be represented as

$$N(i) \triangleq f_i(s(i), s(i_1), \dots, s(i_k))$$

where  $s(i)$  represents the driving-gate size of net  $i$ . Nets  $i_1, \dots, i_k$  are the nets that couple to net  $i$  and  $s(i_1), \dots, s(i_k)$  represent their driving-gate sizes respectively.

We define  $T_S$  as the weighted-sum of all the gate-sizes in a circuit, for some given weight vector  $W$  consisting of non negative weights  $w(i)$ , such that  $W = (i : 0 \leq i < n : w(i))$ . Formally, this can be expressed as

$$T_S \triangleq \sum_{i=0}^{n-1} \{w(i) \cdot s(i)\} = (+i : 0 \leq i < n : w(i) \cdot s(i))$$

Gate-sizes can be represented as a gate-size vector  $S$  as following

$$S \triangleq (i : 0 \leq i < n : s(i))$$

The objective of the gate-sizing problem is to find the gate-size vector  $S$  which yields minimal weighted-sum  $T_S$  under constraints that the noise  $N(i)$  on every net  $i$  is at most  $U(i)$  and  $l(i) \leq s(i) \leq u(i)$ . Here  $U(i)$  represents the maximum noise net  $i$  can tolerate, while  $l(i)$  and  $u(i)$  represent minimum and maximum driving gate-sizes of that net given by physical and timing constraints. The optimal gate-sizing problem can thus be expressed as finding  $S = (i : 0 \leq i < n : s(i))$  such that

$$\min (+i : 0 \leq i < n : w(i) \cdot s(i))$$

s.t.

$$(\forall i : 0 \leq i < n : (U(i) \geq N(i)) \wedge (l(i) \leq s(i) \leq u(i)))$$

Formally this can be expressed as

$$\begin{aligned} & (\min S : (\forall i : 0 \leq i < n : U(i) \geq N(i)) \\ & \wedge (\forall i : 0 \leq i < n : l(i) \leq s(i) \leq u(i)) : T_S) \end{aligned} \quad (1)$$

Physically it translates to determining a gate-size vector  $S$  which minimizes the total weighted sum of the gate-sizes. If we consider  $T_S$  as an area component, the gate-sizing problem can be translated to finding the minimum area of the gates such that there is no coupling-noise in the circuit. Similarly, total power consumption can be expressed as  $T_P = \sum_{i=0}^{n-1} \{p(i) \cdot s(i)\}$  for some non-negative constants  $p(0), p(1), \dots, p(n-1)$ . The gate-sizing problem thus attains to find a gate-vector which minimizes area and power, while ensuring that coupling-noise on every net in the circuit is at most equal to the maximal noise it can tolerate, and all gate-sizes are within their respective size constraints.

## 3. GATE SIZING AS A FIXPOINT COMPUTATION

### 3.1 Obtaining a fixpoint of the Gate Sizing Transformation

Irrespective of the noise model being used, for any net  $i$ , the noise function  $f_i(s(i), s(i_1), s(i_2), \dots, s(i_k))$  must be monotonically non-increasing on  $s(i)$  and monotonically non-decreasing on  $s(i_j)$  for any  $1 \leq j \leq k$ . Formally, it means the following

$$s(i) < s'(i) \quad (2)$$

$$\Rightarrow f_i(s(i), s(i_1), \dots, s(i_k)) \geq f_i(s'(i), s(i_1), \dots, s(i_k))$$

$$s(i_j) < s'(i_j) \quad (3)$$

$$\Rightarrow f_i(s(i), \dots, s(i_j), \dots, s(i_k)) \leq f_i(s(i), \dots, s'(i_j), \dots, s(i_k))$$

Based on the monotonicity property, we define  $g_i$  to be a function that gives the minimal driving gate-size of a net satisfying noise and size constraints, under given driving

gate-sizes of coupled nets.

$$g_i(s(i_1), s(i_2), \dots, s(i_k)) \triangleq \quad (4)$$

(min  $x : (f_i(x, s(i_1), \dots, s(i_k)) \leq U(i)) \wedge (l(i) \leq x \leq u(i)) : x$ )

A system of equations is thus formed when gate-sizes for all nets are combined:

$$(\forall i : 0 \leq i < n : s(i) = g_i(s(i_1), s(i_2), \dots, s(i_k)))$$

If the gate sizes are represented as a vector  $S = (i : 0 \leq i < n : s(i))$ , and  $G = (i : 0 \leq i < n : g_i)$  is used to represent the vector transformation, the above equation can be written as:

$$S = G(S) \quad (5)$$

A solution to Equation (5) is a gate-size vector  $S$ , which is called a fixpoint of  $G$ . It is shown in the following theorem that Equation (5) is a necessary condition for the solution to the optimal gate-sizing problem.

**THEOREM 1.** *The solution to (1) lies within the solution space of Equation (5)*

**PROOF.** We shall establish by contradiction that the solution to the optimal gate-sizing problem (1) is also a fixpoint of  $G$ . Let  $S'$  be the solution to (1) and not be a fixpoint of  $G$ , that is  $S' \neq G(S')$ . We apply the transformation  $G$  on  $S'$  such that:

$$S'' = G(S')$$

Transformation  $G$  on a vector  $S_1$  which satisfies the upper noise-bound constraint ( $\forall i : 0 \leq i < n : N(i) \leq U(i)$ ) and size-bound constraint ( $\forall i : 0 \leq i < n : l(i) \leq s(i) \leq u(i)$ ), will yield a vector  $S_2$  such that  $S_2 \leq S_1$ , that is ( $\forall i : 0 \leq i < n : s_2(i) \leq s_1(i)$ ). This can be seen from its definition in Equation (4), which ensures that a gate is never over-sized. Since  $S'$  is the solution to the gate-sizing problem, it satisfies the upper noise-bound and size-constraints. Thus  $S'' \leq S'$ .

If  $S''$  is less than  $S'$ , then  $T_{S''} < T_{S'}$ , which cannot be true since the solution to the gate-sizing problem  $S'$  yields the minimal  $T_S$ . This implies  $S'' = S'$  or that  $S' = G(S')$ , which is a contradiction.  $\square$

We thus establish that a solution to the optimal gate-sizing problem is also a fixpoint of  $G$ . However, the converse is not true, that is, an arbitrary fixpoint of  $G$  may not be a minimal solution to (1). In other words, a gate-size vector which satisfies (5) can be a solution to (1) if and only if it yields the minimal weighted gate-size sum among all other fixpoints of  $G$ . We thus need to study the fixpoint of  $G$  which is a solution to the gate-sizing problem and also the way to find it.

A *partial order* is defined as follows. Given a set  $f$ , a binary relation  $R \subseteq f \times f$  is called a partial order if it satisfies the following three conditions:

- reflexive:  $xRx$
- antisymmetric:  $xRy \wedge yRx \rightarrow x = y$
- transitive:  $xRy \wedge yRz \rightarrow xRz$

Consider two gate-size vectors  $X$  and  $Y$ . We define a partial order over the solutions space to the Equation (5) as follows.

**DEFINITION 1.** *We say that  $X = (i : 0 \leq i < n : x(i)) \leq Y = (i : 0 \leq i < n : y(i))$  if and only if ( $\forall i : 0 \leq i < n : x(i) \leq y(i)$ ).*

**THEOREM 2.** *For any  $S_1 \leq S_2$ , we have  $G(S_1) \leq G(S_2)$ .*

**PROOF.** For two gate-size vectors  $S_1 = (i : 0 \leq i < n : s_1(i))$  and  $S_2 = (i : 0 \leq i < n : s_2(i))$ , given  $S_1 \leq S_2$ , from Definition (1) we can state that

$$(\forall i : 0 \leq i < n : s_1(i) \leq s_2(i)) \quad (6)$$

We prove the above theorem using contradiction. If  $G(S_1) \not\leq G(S_2)$  we must have

$$(\exists j : 0 \leq j < n : \quad (7)$$

$$g_j(s_1(j_1), s_1(j_2), \dots, s_1(j_k)) > g_j(s_2(j_1), s_2(j_2), \dots, s_2(j_k)))$$

If we denote

$$m \triangleq g_j(s_1(j_1), s_1(j_2), \dots, s_1(j_k))$$

$$n \triangleq g_j(s_2(j_1), s_2(j_2), \dots, s_2(j_k))$$

The above inequality (7) can now be expressed as

$$m > n \quad (8)$$

Let us now consider the following definitions

$$N_1 \triangleq f_j(n, s_1(j_1), s_1(j_2), \dots, s_1(j_k))$$

$$N_2 \triangleq f_j(n, s_2(j_1), s_2(j_2), \dots, s_2(j_k))$$

Given the monotonicity property of the noise function  $f_j$  from Relation (3), and that  $S_1 \leq S_2$  from Relation(6), we must have

$$N_1 \leq N_2 \quad (9)$$

From the definition of  $n$  and definition of  $g_i$  in (4)

$$\begin{aligned} n &= g_j(s_2(j_1), s_2(j_2), \dots, s_2(j_k)) \\ \Rightarrow f_j(n, s_2(j_1), s_2(j_2), \dots) &\leq U(j) \\ \Rightarrow N_2 &\leq U(j) \end{aligned}$$

Combining Relation (9) and the one obtained above

$$\begin{aligned} (N_1 \leq N_2) \wedge (N_2 \leq U(j)) \\ \Rightarrow N_1 &\leq U(j) \\ \Rightarrow f_j(n, s_1(j_1), s_1(j_2), \dots) &\leq U(j) \end{aligned}$$

However, from the definition of  $m$  and  $g_i$

$$\begin{aligned} m &= g_j(s_1(j_1), s_1(j_2), \dots, s_1(j_k)) \\ \Rightarrow m &= \min \{x : (f_j(x, s(j_1), s(j_2), \dots) \leq U(j))\} \\ \Rightarrow (\forall k : f_j(k, s(j_1), s(j_2), \dots) &\leq U(j) : m \leq k) \\ \Rightarrow m &\leq n \end{aligned}$$

This contradicts (8).  $G$  is thus shown to be a monotonic and a convergent transformation, such that  $G(S_1) \leq G(S_2)$  for  $S_1 \leq S_2$ .  $\square$

Theorem(2) shows that  $G$  is a monotonic transformation with respect to the partial order  $\leq$  defined on the gate-size

vectors. According to lattice theory [5], a partially ordered set forms a *complete lattice* if it has a least upper bound and a greatest lower bound on any subset of its elements. We define our greatest lower bound by the point where all gate-sizes are equal to their individual lower bounds given by physical and timing constraints. For our partial ordered set to be a complete lattice, we need to define a virtual upper bound on the elements. We define our virtual upper bound as the point which is reached when any gate-size in the gate-size vector exceeds its upper bound given by physical and timing constraints. It is a virtual point as it is actually a mapping of all points in the partial ordered set with at least one upper size-bound violation. This makes the family of gate-size vectors with the  $\leq$  relation a complete lattice.

Given a subset  $A$  of elements in a complete lattice  $L$ , we use  $\bigvee A$  and  $\bigwedge A$  to represent the least upper bound and the greatest lower bound of elements in  $A$ , respectively. The existence of a fixpoint in our system is guaranteed by the following theorem due to Knaster and Tarski [5].

**THEOREM 3 (KNASTER-TARSKI).** *Let  $L$  be a complete lattice and  $G : L \rightarrow L$  an order-preserving map. Then*

$$\bigvee \{x \in L : x \leq G(x)\} \in \text{fix}(G),$$

where  $\text{fix}(G)$  is the set of fixpoints of  $G$ .

The above theorem is however, not used to compute a fixpoint since it is not feasible to compute the set  $\{x \in L : x \leq G(x)\}$ . Furthermore, we do not know whether such fixpoint is also a smallest solution to the gate-sizing problem. People usually use the iterative method (also called *successive approximation*) to find a fixpoint. That is, they first select an initial solution  $X_0$  and iteratively compute  $X_1 = G(X_0), X_2 = G(X_1), \dots$  in the hope of finding an  $X_n$  such that  $X_n = G(X_n)$ . But the hope can not be fulfilled by starting from any initial point. Fortunately the bottom and the top elements are good candidates for that.

Following a tradition in lattice theory, we use  $\perp$  to represent the bottom element of our complete lattice. We have  $\perp = \{l(i_1), l(i_2), \dots, l(i_n)\}$ , where  $l(i)$  denotes the minimum gate-size for the driving gate of net  $i$  as defined earlier. Since  $\perp \leq G(\perp)$ , based on the monotonicity of  $G$ , we have an ascending chain  $\perp \leq G(\perp) \leq G^2(\perp) \leq \dots$ . If the chain has only finite elements, which is true on any finite solution space, the process will finally reach a fixpoint. Actually, the only property we use of  $\perp$  is  $\perp \leq G(\perp)$ . Therefore, any solution  $X_0$  such that  $X_0 \leq G(X_0)$  can be used as an initial solution to reach a fixpoint. We need to find the least fixpoint, and this can be done by starting with  $\perp$  element. The solution obtained will be the optimal one if the least fixpoint reached is indeed a lower bound on all fixpoints of  $G$ .

**THEOREM 4.** *If  $\text{fix}(G) = \{S_{f_1}, S_{f_2}, \dots, S_{f_l}\}$  denote the set of fixpoints of  $G$ , then there exists a lower bound fixpoint  $S_{f_L} \in \text{fix}(G)$  called the least fixpoint, such that  $(\forall j : 0 \leq j \leq l : S_{f_L} \leq S_{f_j})$ .*

**PROOF.** We define the least fixpoint as:

$$S_{f_L} \triangleq (i : 0 \leq i < n : s_{f_L}(i)), \quad s_{f_L}(i) = \min(s_{f_1}(i), \dots, s_{f_l}(i))$$

Gate-size vector  $S_{f_L}$  is thus the lower bound on all the fixpoints, but we need to show that  $S_{f_L} \in \text{fix}(G)$  to prove our claim. From the monotonicity properties of the noise functions in Relations (4) and (9), we know that sizing down

driver gate-sizes of nets coupled to a net reduces coupling-noise on that net. However, the driving gate-size of a net  $k$  in consideration could have been sized down, and we need to ensure that the noise on the net is still under its upper noise bound. From the definition of  $S_{f_L}$ , there must exist a fixpoint  $S_{f_j}$  with the same driving gate-size for net  $k$ , such that  $s_{f_L}(k) = s_{f_j}(k)$  and  $(\forall i : 0 \leq i < n, i \neq k : s_{f_j}(i) \geq s_{f_L}(i))$ . Since  $S_{f_j}$  is a fixpoint, the upper noise-bound condition is satisfied. Thus, given the monotonicity properties of noise functions,  $S_{f_L}$  must also satisfy the upper noise-bound condition. Additionally since it is the lower bound on all the fixpoints, transformation  $G$  on  $S_{f_L}$  will yield  $S_{f_L}$ . It is thus a fixpoint of  $G$ , and is called the least fixpoint.  $\square$

**COROLLARY 4.1.** *The fixpoint reached starting from the  $\perp$  element of the lattice is the least fixpoint and is the optimal solution to the gate-sizing problem, provided the fixpoint  $\neq \top$ , which implies no solution.*

The least fixpoint reached is the lower bound on the gate-size vectors that are fixpoints of  $G$ , and thus for positive weights  $w(i)$  will yield the minimal  $T_S$ . It is thus the solution to the gate-sizing problem as defined in (1). Notice that the least fixpoint does not depend on the given weight vector  $W = (i : 0 \leq i < n : w(i))$ . This means that the solution is the optimal solution for all non-negative weights.

**COROLLARY 4.2.** *The solution to the gate-sizing problem found above is also a solution to the following problem:*

$$( \min S : (\forall i : 0 \leq i < n : U(i) \geq N(i)) \wedge (\forall i : 0 \leq i < n : l(i) \leq s(i) \leq u(i)) : S = (i : 0 \leq i < n : s(i)) )$$

This means that the solution to the gate-sizing problem is a gate-size vector  $S$ , which gives the smallest possible size of individual gates in the circuit, such that coupling-noise on every net is at most equal to the maximal noise it can tolerate, under given gate-size constraints.

## 3.2 Scheme of Chaotic Iterations

Before we design a good iterative order for updates, we need to establish its theoretical validity. The scheme of *chaotic iteration* [4] ensures that the process will always converge to the same fixpoint, irrespective of the order being used. Transformation  $G$  is composed of a set of partial transformations  $g_1, g_2, \dots, g_n$ . In each step, one or more partial transformations are applied to update gate-size information on some points. Gate-sizes on all other points are kept the same. We will use  $G_A$  to represent such a partial transformation done in one step, where  $A$  represents the points where gate-sizes are updated.

**LEMMA 1.** *If  $X \leq G(X)$ , then  $X \leq G_A(X) \leq G(X)$ ; if  $X \geq G(X)$ , then  $X \geq G_A(X) \geq G(X)$ .*

The above lemma states that no matter what evaluation order is used, the generated sequence is monotonic in the same direction and it will not over-shoot the fixpoint generated by  $G$ . Furthermore, if the evaluation order is fair, that is, a partial transformation will always be applied if its inputs and outputs are not consistent, then the chaotic iteration will always reach the same fixpoint as  $G$ .

## 4. ITERATIVE SIZING ALGORITHM

We propose the following iterative algorithm. Layout extraction is performed on a given circuit and is used to construct the corresponding coupling-graph. We set the driver-size of each net corresponding to the nodes in the coupling-graph to its respective lower size-bound initially, that is

Circuit	# of nets	# of edges	# of Initial Violations	# of Final Violations		Noise Reduction		Run Times(s)	
				List Tr.	Queue Tr.	List Tr.	Queue Tr.	List Tr.	Queue Tr.
circuit_1	20K	60K	28	0	0	100%	100%	0.47	0.11
circuit_2	32K	100K	57	0	0	100%	100%	0.50	0.19
circuit_3	40K	130K	87	0	0	100%	100%	0.50	0.30
circuit_4	166K	450K	579	7	7	98%	98%	2.66	1.09

**Table 1: Noise Reduction Results.**  $U_1(i)$  = Upper Noise Bound for net  $i$

( $\forall i : 0 \leq i < n : s(i) = l(i)$ ). Iteratively we perform updates until noise violations are eliminated from every node, or it is found that gate-sizing cannot remove all violations. In the latter case, no solution is declared. For the updates, the noise on a node  $i$  is calculated based on the noise-model used. Nodes that have a direct edge to node  $i$  induce coupling-noise on node  $i$  and are used to determine the value of  $N(i)$ . If the noise  $N(i)$  exceeds  $U(i)$ , the driver-gate is sized up such that  $N(i)$  equals  $U(i)$ . No update is performed if the calculated noise is less than  $U(i)$ . If the sizing up violates constraints, no solution for complete noise-elimination is declared. We may optionally continue iterations to determine the most number of violations that could be eliminated. The pseudo-code of the proposed algorithm is shown in Figure 2.

<ul style="list-style-type: none"> <li>• Algorithm: Post-route optimal driver-sizing</li> <li>• <b>Input:</b> Layout extraction results</li> <li>• <b>Output:</b> Optimal gate-sizes, if solution exists</li> <li>• begin           <ol style="list-style-type: none"> <li>1. construct coupling-graph <math>G</math> based on layout extraction</li> <li>2. initialize all driver-sizes to their minimum (<math>\forall i : 0 \leq i &lt; n : s(i) = l(i)</math>)</li> <li>3. while (Noise constraint not met for all nodes <math>\wedge</math> each gate-size <math>\leq</math> its upper bound (<math>\forall i : s(i) \leq u(i)</math>))               <ol style="list-style-type: none"> <li>4. for each node <math>i</math> in <math>G</math> with noise violation</li> <li>5. <math>s(i) = g_i(s(i_1), s(i_2), \dots, s(i_k))</math></li> </ol> </li> </ol> </li> <li>• end</li> </ul>
---

**Figure 2: Post-route optimal driver-sizing algorithm**

The algorithm is independent of the weights  $w(i)$  in the gate-sizing problem for non-negative weights. As mentioned above, the order of the iterations may only alter the rate of convergence, but we are certain to reach the optimal solution in any case, provided it exists. We present two possible iterative schemes for node updates below:

#### 4.1 List Traversal

We maintain a list of all nodes and perform updates on the nodes along the list. The iteration restarts when the end of the list is reached and is started all over again. If no updates were found necessary during the entire list traversal, future iterations are stopped and the solution is presented. At any moment if a driver size exceeds given constraints, iteration is stopped and no solution for the circuit is declared. Optionally that node is ignored in future iterations and the algorithm tries to eliminate noise violations on other nodes.

In either case, no solution to complete coupling-noise elimination by gate-sizing is declared.

#### 4.2 Queue Traversal

This scheme involves using a queue which is filled with nodes having initial noise violations. As a node  $i$  is popped from the queue, its driver is sized up to eliminate its noise violation. All nodes having an edge from node  $i$  are pushed in the queue, if they have noise violations and are not already in the queue. Iteration stops either when the queue is empty and we have a solution, or when a driver size exceeds the given constraint and no solution is concluded. As in List Traversal, we may optionally go ahead and try removing other violations.

### 5. RESULTS

We present results of our algorithm on four circuits, namely *circuit\_1*, *circuit\_2*, *circuit\_3*, and *circuit\_4* respectively. Due to unavailability of benchmarks used by Becer et al. [2], the circuits were randomly generated with realistic parameters in a  $0.18\mu\text{m}$  technology. We have used the  $2\pi$  model [3] as the noise-model for our simulations. For the test circuits, the driver resistance  $R_d$  is from 20 to  $2000\Omega$ , loading capacitance  $C_l$  is from 4 to  $50\text{fF}$ , and the slew is from 10 to 300ps. Typically gate-sizes were bounded in each direction by a factor of at most 2.

Circuit	# of nets	# of Initial Violations	# of Final Violations	Noise Reduction
circuit_1	20K	277	11	96%
circuit_2	32K	524	19	96%
circuit_3	40K	1425	63	95%
circuit_4	166K	1356	30	97%

**Table 2: Noise Reduction Results, using QT with tighter Upper Noise Bound  $U_2$  : Avg  $\frac{U_1(i)}{U_2(i)} \approx 1.3$**

Circuit	# of nets	# of Initial Violations	# of Final Violations	Noise Reduction
circuit_1	20K	755	54	92%
circuit_2	32K	1447	90	93%
circuit_3	40K	3332	316	90%
circuit_4	166K	4040	244	93%

**Table 3: Noise Reduction Results, using QT with tighter Upper Noise Bound  $U_3$  : Avg  $\frac{U_1(i)}{U_3(i)} \approx 1.6$**

In Table (1) we present the results of our simulation using both the List Traversal (LT) and the Queue Traversal (QT) techniques. The upper noise bound  $U_1(i)$  on the nets were set to simulate initial number of violations like the ones used in the benchmarks from [2]. We show the # of nets, the # of edges in the coupling-graph formed, the total

Circuit	# of nets	# of Initial Violations	# of Final Violations	Noise Reduction
circuit_1	20K	1975	291	85%
circuit_2	32K	3558	571	83%
circuit_3	40K	7324	1599	78%
circuit_4	166K	11374	1380	87%

**Table 4: Noise Reduction Results, using QT with tighter Upper Noise Bound  $U_4$  :  $\text{Avg } \frac{U_1(i)}{U_4(i)} \approx 2.0$**

Circuit	# of nets	# of Initial Violations	# of Final Violations	Noise Reduction
circuit_1	20K	4560	1397	69%
circuit_2	32K	7947	2597	67%
circuit_3	40K	14256	5898	58%
circuit_4	166K	29776	7571	74%

**Table 5: Noise Reduction Results, using QT with tighter Upper Noise Bound  $U_5$  :  $\text{Avg } \frac{U_1(i)}{U_5(i)} \approx 2.7$**

number of violations before and after the optimization, % of Noise Reduction and run times. It is to be noted that the *Initial Number of Violations* in the tables refer to the violations existing in the original circuit and not the number of violations after all gate-sizes have been set to their lower bound. Table (2) to Table (6) show results of similar simulations using the Queue Traversal, but with progressively tightened upper noise-bound constraints. It is observed that the number of failing nets increase as the upper-noise bound is constrained progressively. The Noise Reduction % shown in the tables do not reflect the maximal noise reduction, since the algorithm simply neglects the node and continues. In this case, it reports *No Solution* to noise elimination by driver-sizing, and just reports the percent of reduction when it exits, which is not necessarily the maximal. The % thus reflects the worst case reduction using the algorithm.

Figure 3 shows coupling-graphs before and after the algorithm is run on a tiny test circuit. Here two directed edges in either direction between two nodes represent an undirected edge. The coupling-graph has been generated using XVCG [8] which has been integrated with the simulation environment. The simulations have been run on a Pentium 2.4GHz Xeon processor server, having 1GB RAM and running RedHat Linux 8.0.

## 6. FUTURE WORK

In this paper, we presented a method to achieve the optimal solution to the gate-sizing problem. However if the optimal solution does not exist, the present algorithm does not return the least number of violations, since the *Scheme of Chaotic iterations* assumes that the optimal solution exists. We plan to extend this work to determine the minimal number of violations that exist, if there is no optimal solution to the initial problem.

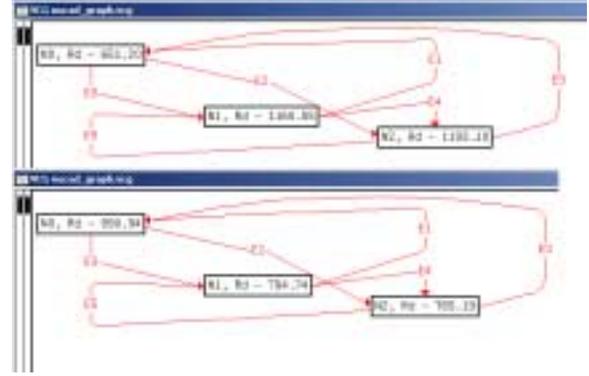
Currently we have considered gate-sizes to be bound by timing and physical constraints. This may lead to an worst case prediction. We consider working on the problem with path-delays as constraints in the future, so that individual gate-sizes are bound only by physical constraints.

## 7. ACKNOWLEDGEMENT

This research was supported by the National Science Foundation under grant CCR-0238484.

Circuit	# of nets	# of Initial Violations	# of Final Violations	Noise Reduction
circuit_1	20K	9330	5158	44%
circuit_2	32K	15538	8974	42%
circuit_3	40K	24020	15781	34%
circuit_4	166K	66334	33623	49%

**Table 6: Noise Reduction Results, using QT with tighter Upper Noise Bound  $U_6$  :  $\text{Avg } \frac{U_1(i)}{U_6(i)} \approx 4.0$**



**Figure 3: Coupling graph of a circuit generated by XVCG**

## 8. REFERENCES

- [1] Semiconductor Industry Association. National technology roadmap for semiconductors, 1997.
- [2] M. R. Becer, D. Blauuw, I. Algor, R. Panda, C. Oh, V. Zolotov, and I. N. Hajj. Post-Route Gate Sizing for Crosstalk Noise Reduction. In *Proc. of the Design Automation Conf.*, pages 954–957, 2003.
- [3] J. Cong, D. Z. Pang, and P. V. Srinivas. Improved Crosstalk Modeling for Noise Constrained Interconnect Optimization. In *ACM Intl. Workshop on Timing Issues in the Specification and Synthesis of Digital Systems*, 2000.
- [4] P. Cousot and R. Cousot. Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *ACM Symposium on Principles of Programming Languages*, pages 238–252, Los Angeles, CA, January 1977.
- [5] B. A. Davey and H. A. Priestley. *Introduction to Lattices and Order*. Cambridge, 1990.
- [6] D. Gries and F. B. Schneider. *A Logical Approach to Discrete Math*. Springer-Verlag New York, Inc., 1994.
- [7] M. Hashimoto, M. Takahashi, and H. Onodera. Crosstalk noise optimization by post-layout transistor sizing. In *International Symposium on Physical Design*, pages 126–130, 2002.
- [8] I. Lemke and G. Sander. Visualization of Compiler Graphs. In *Design report, USAAR-1025-visual, ESPRIT Project 5399 Compare, Universität Saarlandes, FB 14 Informatik*, pages D 3.12.1–1, 1993.
- [9] T. Xiao and M. Marek-Sadowska. Gate sizing to eliminate crosstalk induced timing violation. In *Proc. Intl. Conf. on Computer Design*, pages 186–191, 2001.
- [10] H. Zhou. Timing analysis with crosstalk is a fixpoint on a complete lattice. In *IEEE Transactions on Computer-Aided Design, September 2003*, pages 1261–1269.