Crosstalk Reduction by Transistor Sizing*

Tong Xiao and Malgorzata Marek-Sadowska
Electrical and Computer Engineering Department
University of California, Santa Barbara, CA 93106, USA

Abstract

In this paper we consider transistor sizing to reduce crosstalk. First, crosstalk noise dependency on wire width, wire spacing, driver and receiver sizes are discussed, and validated by experiments. Then transistor sizing for timing and noise is discussed and solved using optimization techniques. Experimental results suggest that crosstalk violations can be removed by transistor sizing with very small area ovehead.

I. Introduction

With aggressive technology scaling, crosstalk effects on chip level cannot be ignored. It is so because line to line coupling increases when line to line spacing decreases and the aspect ratio increases, as it is the case in deep submicron technologies. Crosstalk can cause delay faults or logic faults [1]. The existing techniques employed to reduce crosstalk noise or coupling effect on delay fall into two categories:

1. Parallel runs of two adjacent lines are restricted not to exceed a certain limit. Using this criterion, post-global routing optimization[16] or wire ordering in detail routing stage [6] have been proposed.

2. Spacing between neighboring lines is increased to reduce noise and delay caused by coupling [3].

The above approaches cannot guarantee that crosstalk noise will be reduced below a desired limit. Using coupling length or coupling capacitance as sole criterion can either over-constrain a layout tool or be too optimistic in some cases.

This paper targets crosstalk noise directly, not coupling length or coupling capacitance. First, the dependency of crosstalk noise on coupling length, wire width, spacing, and transistor sizes are discussed, and it is shown that transistor sizing and spacing are the most effective ways to reduce crosstalk noise for a given layout. Transistor sizing as a circuit technique to control crosstalk was mentioned for the first time in [14] and [15]. This is the first paper in literature that discusses transistor sizing to reduce crosstalk noise while satisfying timing constraints. Crosstalk noise is very sensitive to driver sizes, it is possible that a crosstalk free layout may have crosstalk violations after its transistors have been sized for timing.

In section III of this paper, the optimization problem of transistor sizing for timing and noise is discussed. In section IV two approaches to solve transistor sizing for timing and noise are proposed. It is shown that the area overhead due to transistor sizing for noise reduction is small, and noise violation after timing optimization can be efficiently eliminated. Section V concludes the paper.

II. Preliminaries

When a quiet line is coupled to one or more switching lines, a noise signal may appear on it. If coupling is strong enough, the quiet line may switch and cause a wrong logic operation. The purpose of crosstalk reduction is to make sure that coupling will not cause wrong logic operation in the circuit. Coupling may also speed up lines which switch in same direction, or slow down lines switching in opposite directions comparing to the case when a switching line is coupled to some quiet lines[9].

Foundation 1 The parallel runs length of adjacent wires is not a correct criterion to decide if a wire is noise critical or not. This is because crosstalk noise depends on many factors, and coupling length has a week correlation to noise. Measurements and simulation[2] show that when parallel length changes from 1mm to 10mm in 0.35 μm CMOS technology, noise on a victim wire doesn't change much.

A simple yet effective model which clearly reflects the impact of driver sizing and other physical dependencies has been proposed in [14]. It has been extended to include wire resistance and is applicable to multiple coupling cases in [15]. Peak noise is given by:

$$V_p^A = \frac{\sum_{X_i \in X} X_i \cdot \Re_i}{\sum_{C_i \in C} C_i \cdot R_{ii}} \qquad (1)$$

We will use the above formula (1) derived in [15] to estimate crosstalk noise. It is an upper bound of 20% accuracy comparing with HSPICE simulation. In (1) X is the set of all coupling capacitance, and C is the set of all capacitance in the circuit. R_i is the driving point resistance seen from node i (where X_i is connected) with all capacitance open, while R_{ii} is the resistance across the leads of the ith capacitor.

Foundation 2 The impact of different wire sizes on crosstalk noise is much smaller comparing to the impact of wire spacing, thus spacing is more effective to use when doing noise reduction.

Table 1 shows crosstalk noise on quiet line_1 of width w1 which runs in parallel to a switching line_2 of width w2. Both wires are 1mm long and their widths are in μm. All gates are minimum sized, and commercial extraction tool was used to get line resistance and capacitance. Transistor model for 0.35 technology is used in HSPICE simulation. The

Table 1: Effect of wire sizing and spacing

w2 \ w1	0.4	0.6	0.8	1.0
0.4	1/1	0.93/0.76	0.87/0.61	0.81/0.50
0.6	0.94/0.76	0.87/0.65	0.81/0.49	0.77/0.417
0.8	0.88/0.65	0.83/0.49	0.78/0.41	0.74/0.35
1.0	0.85/0.49	0.80/0.41	0.76/0.35	0.72/0.30

*This work was supported by National Science Foundation grant MIP 94119 and the California Micro Program through Mentor Graphics and Rockwell.

results are normalized to the noise on line_1 when minimum wire sizes and spacing are used.

In table's entries, the first data is the normalized noise when wire sizes have widths of w1 and w2 respectively, the second data is the normalized noise when minimum width wire sizes are used, and only spacing is increased such that total overhead by spacing and wire sizing are the same. For example, when w1=0.6μm and w2=0.8μm, the corresponding spacing among minimum width wires is 1.0μm, as both cases have same total area 1.8μm. In row 1 spacing is changed from 0.4 to 1.0 by 0.2 each time; in row 2, spacing is changed from 0.6 to 1.2 by 0.2 each time, and so on.

Increasing wire size of either line can reduce crosstalk noise on line_1, but the change is much smaller than in the corresponding spacing case. For example, when w1=0.8μm, w2=0.6μm, noise is reduced from 1 to 0.81, but spacing only can reduce noise to 0.49.

Foundation 3 Increasing driver size of a wire, while keeping everything else unchanged, has big impact on coupling noise. The impact is twofold: the noise on the driven wire decreases but increases on its neighbors. Measurement and simulation in [2] supported this claim.

Foundation 4 Increasing size of a wire's receiver, while keeping everything else unchanged, will decrease noise on the wire, but the impact is quite small comparing to changing driver's size.

Table 2 shows the normalized noise on line_1 under different conditions. Two parallel lines of length 1 mm are considered. Noise is normalized to the value when the drivers and receivers are minimum sized. In row 1, the driver on line_1 is minimum size, twice minimum size, up to 5 times minimum size. The first data in each entry is the normalized noise on line_1 when line_2 is switching, the second data is noise on line_2 when line_1 is switching. Increasing a driver size and keeping everything else unchanged can dramatically improve the net's noise immunity, but it will make the neighbor suffer bigger noise.

Table 2: Effect of driver and receiver sizing

	1	2	3	4	5
1 driver	1/1	0.78/1.2	0.59/1.3	0.46/1.4	0.36/1.4
1 receiver	1/1	0.99/1.0	0.99/1.0	0.99/1.0	0.98/1.0
2 drivers	1	1.05	1.02	0.98	0.97

In row 2 of Table 2, the receiver on line_1 is minimum size, twice minimum size, up to 5 times minimum size. The impact on noise on line 1 and line 2 is rather small. In row 3, both drivers on line 1 and line 2 are changed to the same size. In this case the noise changes are minor.

Foundation 5 Increasing spacing between two parallel lines reduces coupling capacitance, crosstalk noise and coupling-caused delay.

For example, in 0.35 μm technology, increasing spacing from 0.4 μm to 0.6 μm between 1 mm long lines reduces noise on the quiet line by 25% (see table 1).

Foundation 6 In some cases, transistor sizing alone may not rectify all noise violations. In such a case spacing between some wires has to be increased (if no rerouting or buffer insertion are applied).

For example, such a case occurs when 2 long lines are adjacent to each other and both are noise-critical.

Foundation 7 Transistor sizing for timing optimization or timing-constrained optimization may make non-noise critical wires become critical.

Foundation 8 When using Elmore delay model to calculate delay, coupling capacitance can be treated as effective ground capacitance. Due to Miller effect, the effective ground capacitance of a coupling capacitance Cc can be 0, Cc or 2Cc, depending on the switching direction on victim and aggressor wires [9].

III. Problem formulation and analysis

The crosstalk aware transistor sizing problem can be formulated as:

minimize $Area$

subject to $Delay \le Tspec$

and $Crosstalk - noise \le Nspec$

and $each - transistor - size \ge min - channel - width$

where Tspec and Nspec are user provided timing constraints or noise constraints. We treat transistor sizing problem as deciding transistor channel widths. The objective function is the summation of all channel widths

$f(x) = \sum_{i=1}^{N} x_i$, where N is total number of transistors.

1. The models

To estimate the peak crosstalk pulse we use formula (1) derived in [15]. Elmore delay model [4] serves as a timing metric where 2Cc to ground replaces the coupling capacitance Cc. From Foundation 9, we know that this is the worst case delay effect of a coupling capacitance.

We apply π model for interconnect segments, make use of the fact that driver's resistance is inversly proportional to transistor's size, and transistor's capacitance is proportional to transistor channel width. It is well know that under this assumptions, Elmore delay is a posynomial function. Transistor sizing for timing is not a new problem. The objective function and timing constraints can be changed to convex functions under appropriate variable transformation, thus an exact solution can be obtained by applying convex programming techniques[12]. A simpler and faster iterative algorithms were used in TILOS[5], and in [13].

2. Nonlinearity of the combined constraints problem

Crosstalk constraints are more complicated than timing constraints. Let's first consider a 2 net coupling case. A circuit model of two parallel lines is shown in fig. 1.

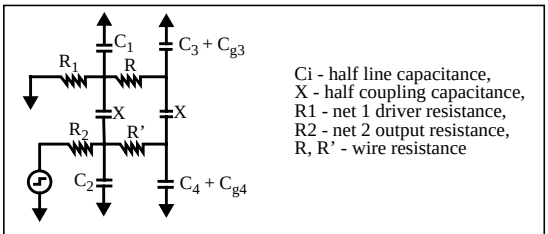

Fig. 1. Circuit model of 2 coupled nets.

The crosstalk constraint for this case is:

$$V^A{}_p \le N_{spec} \quad (2)$$

By applying formula (1), we have 4 variables, i.e., R_1, R_2, C_{g3} and C_{g4}. Each R_i is proportional to $1/x_i$, and each gate capacitance C_{gi} is proportional to x_i, where x_i is the size (channel width) of the ith transistor. So the above constraint can be transformed to:

$$N(x) = \frac{a_1}{x_1} - \frac{a_2}{x_2} - a_3 \cdot x_3 - a_4 \cdot x_4 - a_5 \cdot \frac{x_3}{x_1} - a_6 \cdot \frac{x_4}{x_2} + b_1 \le 0 \quad (3)$$

Where $a_1 = A \cdot \left(\frac{2X}{N_{spec}} - C_1 - C_3 - 2X \right)$,

$a_2 = A \cdot (C_2 + C_4 + 2X)$, $a_3 = B \cdot R$,

$a_4 = B \cdot R'$, $a_5 = a_6 = A \cdot B$,

$b_1 = \frac{R \cdot X}{N_{spec}} - R \cdot C_3 - R' \cdot C_4 - R \cdot X - R' \cdot X$.

And A and B are, respectively, the unit resistance and unit capacitance.

When one more line couples to the victim net, two more variables will appear. For net i, let x_i denotes the size of the driver, and y_is denote the sizes of receivers.

$$N(x) = \frac{a_i}{x_i} - \sum_{j \ne i} \frac{a_j}{x_j} - \sum_{j=1}^{N} c_j \cdot y_j - \sum_{j=1}^{N} d_j \cdot \frac{y_j}{x_j} + b_i \le 0 \quad (4)$$

The noise constraints (4) have quite a regular form. A function f is convex iff its Hessian matrix

$$H(x) = \left[\frac{\partial^2 f}{\partial x_i \partial x_j} \right]_{ij}$$

is positive semidefinite[8]. The function N(x) in (3) is not convex, as for N(x):

$$\frac{\partial^2 N}{\partial x_2^2} = -\left(a_2 + a_6 \cdot x_4 \right) \frac{2}{x_2^3} < 0$$

which makes at least 1 eigenvalue of matrix H(x) negative, so matrix H(x) is not positive semidefinite[11]. For the same reason, even after variable transformation $x_i = e^{z_i}$, the function is not convex either.

We will make two simplifications to get a easier problem. First, we will consider only sizing transistors up. This is a reasonable assumption, as we may be starting from the minimum initial features. Second, in crosstalk constraints, we will ignore the impact of changing gate capacitance by treating each gate capacitance as constant. This is reasonable, as from foundation 4, it follows that the effect of receiver sizing is small when line capacitance is much bigger than receiver's gate capacitance.

Now we can rewrite crosstalk constraints into a simpler form:

$$N(x) = \frac{a_1}{x_1} - \frac{a_2}{x_2} + b_1 \le 0$$

The solution which satisfies this constraint is guaranteed to be a feasible solution to satisfy (3).

Note that now in (1), for each line coupling to a victim net, only one variable will appear. In general, for the case when more than one line couples to a victim net, crosstalk noise constraint has the form of $N(x) = \frac{a_i}{x_i} - \sum_{j \ne i} \frac{a_j}{x_j} + b_i \le 0$.

It is still non convex and cannot be transformed into convex function using the standard transformation $x_i = e^{z_i}$. But it's linear in 1/xi. So far we can see that transistor sizing satisfying timing and crosstalk constraints at the same time is a difficult nonlinear programming problem, it does not fall into any special category.

A practical strategy may be separating timing and crosstalk optimizations. As small changes of transistor sizes may be enough to satisfy crosstalk constraints, we can perform timing optimization first, followed by crosstalk optimization. After timing optimization is done, we use timing budget for each gate in the form of $R_i \le R_{i,max}$, this is the same as only sizing transistors up. We can transform our transistor sizing problem as deciding transistor resistance:

minimize $\sum_{i=1}^{N} \frac{1}{R_i}$ (5)

subject to $R_i \le R_{i,max}$, i = 1, 2,..., N (6)

and $N(x) = a_i \cdot R_i - \sum_{j \ne i} a_j \cdot R_j + b_i \le 0$, i =, 2,..., N (7)

This linearly constrained optimization problem can be solved easily.

IV. Solving the optimization problem

We define a PROFILE of a wire to be a set of all wires that are transitively adjacent to a given wire. Two wires are transitively adjacent if they are adjacent or there exists a sequence of neighbors such that the first one in the sequence is neighbor of the wire in question. Drivers of these wires are called profile drivers.

1. Solving the NLP problem for transistor sizing

We will solve the problem in two different ways:

(1) Solve the NLP problem including simultaneously timing and noise constraints. For timing constraints, we will include timing-critical and some near-critical nets. In the worst case scenario, all wires which are in PROFILEs of the transistors included in timing constraints have to be considered as potential noise critical wires. This introduces a lot of linear constraints, many of which may be spurious as sizing for timing will not affect all wires. Practically solvable problem sizes may be limited.

(2) Solve the problem by iterating transistor sizing for timing and transistor sizing for noise which are easier to solve than combined problem. We start from a crosstalk free starting point. It may be the minimum sized circuit in which crosstalk violations have been rectified by changing spacing. We solve the conventional transistor sizing for timing, then solve (5), (6) and (7), which are the sizing problem when all transistors are only sized up.

After that, verification of the circuit should be done to make sure there are no new timing violations, if there are, return to the iteration loop. If the process fails to converge, spacing should be applied.

2. Experiments

Recently, several solvers for NLP problems have been available on NEOS[7], including MINOS, LANCELOT, and SNOPT, etc. After testing them, we decide to use SNOPT to solve our optimization problem. SNOPT implements a sequential programming algorithm that uses smooth augmented Lagrangian merit function and makes explicit provision for infeasibility in the original problem and for the quadratic programming subproblems.

The results for randomly generated circuits are listed in Table 3. The first column shows the number of nodes in each circuit. Both approaches discussed above have been implemented and their sizing results and run time are compared to the result of sizing transistors for timing constraints only. The second column gives the area overhead when crosstalk noise constraints are included and the problem is solved by NLP. The third column shows the area overhead when iterative method is applied. The fourth column shows the run time increase due to applying NLP over the transistor sizing for timing only. The fifth column shows run time penalty of the iterative approach normalized to transistor sizing for timing. The column labeled "violation" shows the ratio of noise violations with respect to all coupling wires after sizing for timing only. The column labeled "weight" shows the ratio of number of nodes which contribute to the noise constraints to the total number of nodes in the circuit. The last row gives the averages.

TABLE 3. Transistor sizing results

nodes	Pa1	Pa2	Pt1	Pt2	violation	weight
55	0.0034	0.0073	-0.0056	0.0000	0.1429	0.2545
65	0.0088	0.0165	0.1438	0.0016	0.1250	0.2462
67	0.0001	0.0010	0.9363	0.0026	0.1250	0.2388
71	0.0052	0.0144	0.9530	0.0033	0.3333	0.2535
76	0.0025	0.0064	0.2635	0.0031	0.1818	0.2895
78	0.0083	0.0126	0.2978	0.0000	0.2727	0.2821
85	0.0012	0.0034	-0.3990	0.0000	0.1000	0.2353
89	0.0066	0.0105	0.7545	0.0039	0.2500	0.2697
90	0.0012	0.0016	-0.2347	0.0022	0.0909	0.2444
94	0.0002	0.0012	0.5401	0.0021	0.0833	0.2553
100	0.0022	0.0047	-0.1972	0.0014	0.0769	0.2600
103	0.0003	0.0018	0.0997	0.0016	0.0714	0.2718
112	0.0005	0.0019	1.0253	0.0026	0.0667	0.2679
113	0.0004	0.0017	0.0257	0.0023	0.0625	0.2832
115	0.0001	0.0008	0.2149	0.0000	0.0625	0.2783
119	0.0001	0.0009	0.7736	0.0033	0.0625	0.2689
121	0.0016	0.0059	1.1960	0.0030	0.2353	0.2810
124	0.0020	0.0090	0.6250	0.0026	0.2667	0.2419
124	0.0018	0.0078	0.0531	0.0019	0.2222	0.2903
125	0.0002	0.0019	0.1840	0.0012	0.1333	0.2400
131	0.0006	0.0028	-0.3430	0.0018	0.1053	0.2137
137	0.0003	0.0016	0.4518	0.0025	0.0526	0.2774
143	0.0015	0.0046	0.5825	0.0019	0.1053	0.2657
92	0.0020	0.0054	0.2964	0.0019	0.1564	0.2648

As we can see from the table, area overheads to satisfy noise constrains are small, on the average of the order 0.2% to 0.5%. Area overhead is always larger when decoupled, iterative approach is applied, but its running time is on the average much shorter. For all of the cases tried, noise violation after sizing for timing can always be solved without causing new timing problems, this shows that using sizing up as an extra constraint in (5) to (7), although may cause more extra area overhead, is an effective alternative.

V. Discussion and future work

In this paper, we addressed the transistor sizing problem for timing and noise. We suggest that in this case, spacing should be used only to help transistor sizing. Although spacing and transistor sizing can be formulated together into one large NLP problem, it may generate problems that contain too many variables to be solvable for practical instances. We are now investigating practical ways of solving spacing and transistor sizing for timing and noise.

References:

[1] H. B. Bakoglu, "Circuits, Interconnections, and Packaging for VLSI", Addison-Wesley Publishing Co. 1990.

[2] D. H. Cho, Y. S. Eo, N. H. Kim, J. K. Wee, O. K. Kwon, and H. S. Park, "Interconnect Capacitance, Crosstalk, and Signal Delay for 0.35um CMOS Technology", Proc. IEDM '96, pp.619-622.

[3] J. Cong, L. Hei, C. K. Kok and Z.Pan, "Global Interconnect Sizing and spacing with Consideration of Coupling Capacitance", ICCAD 97, pp. 478-485.

[4] W. C. Elmore, "The Transient Response of Damped Linear Networks with Particular Regard to Wideband Amplifiers", Journal of Applied Physics, Volume 19, Jan-Dec. 1948, p55-63.

[5] J. P. Fishburn, A. E. Dunlop, "TILOS: A Posynomial Approach to Transistor Sizing", Proceedings of ICCAD 85, p326-328.

[6] T. Gao, C. Liu, "Minimum Crosstalk Switchbox Routing", Proc. ICCAD 94, page 610-615, 1994.

[7] NEOS guide, http://www-c.mcs.anl.gov/home/otc/Guide.

[8] C. H. Papadimitriou, K. Steiglitz, "Combinatorial Optimization: Algorithms and Complexity", Prentice-Hall, Inc., 1982, p17.

[9] L. Pileggi, "Coping with RC(L) Interconnect Design Headaches", Proc. ICCAD 95, p246-253.

[10] T. Sakurai, "Closed-Form Expression for Interconnection Delay, Coupling, and Crosstalk in VLSI's", IEEE Trans. on Electron Devices, vol. 40, no. 1, 1993, pp. 118-124.

[11] G. Strang, "Linear Algebra and Its Applications", Third edition, Harcourt Brace Jovanovich, Inc., 1988, p339.

[12] S. S. Sapatnekar, "An Exact Solution to the Transistor Sizing Problem for CMOS Circuits Using Convex Optimization", IEEE Trans. on CAD of IC and Systems, vol. 12, no. 11, 1993, p1621-1634.

[13] J. Shyu, A. Sangiovanni-Vincentelli, J. P. Fishburn, "Optimization-Based Transistor Sizing", IEEE Journal of Solid-State Circuits, vol. 23, no. 2, April 1988.

[14] A. Vittal, M. Marek-Sadowska, "Crosstalk Reduction for VLSI", IEEE Trans. on CAD, March, 1997, vol. 16, no. 3, pp. 290-298.

[15] A Vittal, L.H.Chen,M. Marek-Sadowska, K.P.Wang and X. Yang, "Crosstalk in Resistive VLSI Interconnections", to be presented at 12th International Conference on VLSI Design.

[16] T. Xue, E. S. Kuh and D. Wang, "Post Global Routing Crosstalk Risk Estimation and Reduction", Proc. ICCAD 96, pp. 302-309.