

# Wire Swizzling to Reduce Delay Uncertainty Due to Capacitive Coupling \*

Puneet Gupta<sup>†</sup> and Andrew B. Kahng<sup>‡</sup>

<sup>†</sup> Department of ECE, UC San Diego, La Jolla, CA, USA

<sup>‡</sup> Department of CSE and ECE, UC San Diego, La Jolla, CA, USA

puneet@ucsd.edu, abk@ucsd.edu

## Abstract

Reduction of worst-case delay and delay uncertainty due to capacitive coupling is a still unsolved problem in physical design. We describe a *routing only* layout solution - *swizzling* - which reduces worst-case coupling delay for long parallel wires such as in wide on-chip global buses. We understand that swizzling is a folklore in structured-custom design community but we are the first to describe the method and analyze the potential benefits in literature. We give a general method for construction of good swizzling patterns. We also give empirically determined, optimal swizzling patterns for various technology nodes and typical repeater intervals. From our results, we see up to 31.5% reduction in worst-case delay and 34% reduction in delay uncertainty.

## 1 Introduction

With the rapid move to ultra deep sub-micron designs, it is a difficult challenge to ensure integrity of signals as they traverse conductors on a chip. Crosstalk between signals, due to increased capacitive coupling, impacts timing of signals on a chip and causes functional failures and performance degradation. The 2001 ITRS [10] has identified management of crosstalk as a major challenge. With capacitive coupling contributing more than half of the total interconnect delay, many signal integrity problems and large guardbanding overheads result. It is imperative to develop new techniques to reduce crosstalk and the delay uncertainty caused by it.

### 1.1 Related Works

A number of previous works have proposed techniques to control crosstalk-induced noise and delay. Saxena et al. [3] prove that the variation of total crosstalk on a net is monotone or unimodal within a *basic perturbation interval*. Thus, they perturb trunks to vary inter-wire spacing to maximize the minimum slack over all the nets. Jhang et al. [4] perform track segment permutation to minimize crosstalk. They swap segments of different nets but since they do not split the horizontal or vertical segments of the initial route, their approach is essentially the same as the track assignment approach followed in [6]. Vittal et al. [5] present a new crosstalk model which is not as pessimistic as the charge-sharing model; they determine whether a given routing solution will fail due to coupling noise but ignore the issue of coupling delay. Kirkpatrick et al. [6] propose a track

assignment algorithm to minimize channel width while obeying crosstalk constraints on all the nets. The same authors subsequently observe that since all worst-case coupling signal transitions may not be observable at the primary outputs [7], there is no need for the pessimistic assumptions of worst-case analysis. The concept of *digital sensitivity* is introduced to ignore redundant transitions and reduce the number of crosstalk constraints. Gao et al. [8] give a mixed ILP formulation for the track permutation problem for crosstalk minimization. The objective of the ILP is to maximize *minslack* on timing paths.

One observation with respect to spacing-based crosstalk avoidance methods is that they are unaware of downstream area fill synthesis. Area fill (“dummy fill”) insertion for CMP (chemical-mechanical planarization) uniformity in metallization layers adds metal geometries in empty spaces of the layout, depending on various foundry-specific layout density constraints. The addition of dummy fill increases coupling capacitance [13] and, as buffer distances between dummy and actual features decrease, reduces the benefits of spacing wires apart. A second observation is that *repeater staggering* [1, 14] provides an elegant and simple approach to reduce crosstalk. The technique gives good control of delay uncertainty and limits the maximum amount of worst-case coupling (essentially, compensating with best-case coupling), but incurs high costs in terms of layout perturbation, via blockage, and power. Finally, we note that in current layout methodology, use of clean bus routing (as is common in the contexts of PCB and custom layout) has become less attractive because of worst-case coupling effects. Thus, “randomized” routing may be used to route wide buses, even though this is costly in terms of area, skew, predictability and other quality measures. In summary, existing approaches all have drawbacks (overuse of routing resources, vulnerability to fill-induced coupling, poor routing quality, etc.) when it comes to clean bus routing at the top level of the chip hierarchy.

### 1.2 Key Idea - Arrival Time Displacement

In today’s standard *switch factor* [11] or Miller factor [2] based coupling delay analysis, the switch factor is dependent on relative arrival and slew times of the victim and its aggressors. For instance, a zero-coupling configuration can be obtained from a worst-case coupling configuration, by simply inserting a delay element in the beginning of the aggressor line. If the delay of the delay element is greater than the rise/fall time of the victim (assuming otherwise synchronous operation), only the average-case coupling will result. This reduces the coupling and hence the total delay of the victim, but increases the delay of the aggressor. Such an approach is well suited when the victim is a

\*This research was supported in part by the MARCO Gigascale Silicon Research Center and Cadence Design Systems, Inc.

critical net while the aggressor is not critical, and when the rise time of the input signal is small compared to the total delay of the aggressor.

Observe that introduction of a delay element can be realized by adding a *dogleg* to the routing. This allows us to apply the concept of *arrival time displacement* to the routing of long parallel buses. We propose to intentionally and systematically permute (or, *swizzle*) wires so as to misalign the arrival times along the length of the lines. The swizzling will change the switch factors and reduce worst case delays - hence delay uncertainties as well - of *all* wires in a given bus. Ad hoc permutation of wires to reduce worst-case coupling is, to our knowledge, a known technique in industrial folklore [15], but has never been previously mentioned or analyzed in the literature. A similar twisted-bundle structure was proposed in [9] for inductance compensation. Because swizzling is a very local operation, its effects on routing resources in adjacent layers are both small and predictable.

### 1.3 Organization of the Paper

The remainder of this paper is organized as follows. In Section 2, we describe our switch factor based delay modeling approach. Section 3 gives details and formalism behind the idea of swizzling wires to reduce worst-case delay. Experimental results are given in Section 4, and we conclude in Section 5.

## 2 Delay Modeling

The core of our delay analysis is the computation of arrival and slew times at arbitrary points in the interconnect, in the presence of capacitive coupling. Transient analysis using circuit simulation tools (e.g. *HSpice* [12]) is too computationally expensive. For simplicity of analysis and computation speed, we use the Elmore delay model along with a *switch factor* based analysis of capacitive coupling [11]. Elmore delay at position  $x$  in a line of length  $l$  is given by

$$T_{ED}(x) = \frac{RC}{2} \left( \left(1 - \frac{x}{l}\right)^2 \right) + RC_L \frac{x}{l} \quad (1)$$

where  $R, C$  are resistance and capacitance of the entire line and  $C_L$  is the load capacitance at the line end. We model distributed interconnect by  $L$  segments. We use the following terminology.

- $r$ : resistance of each segment.
- $c_g$ : ground capacitance of each segment.
- $c_c$ : coupling capacitance per segment to one nearest neighbor.
- $C_L$ : load capacitance at the line end.
- $SF_k$ : total switch factor (due to one or two nearest neighbors) experienced by the line at the beginning of segment  $k$ .
- $n$ : total number of segments in the line.

Then, the delay of the  $k_{th}$  segment can be calculated as

$$T(k) = \frac{r(c_g + SF_k c_c)}{2} + rc_g(n-k) + rc_c(\sum_{i=k+1}^n SF_i) + rC_L \quad (2)$$

Equation 2<sup>1</sup> assumes that the correct switch factors for each segment are known *a priori*. Switch factors depend on the aggressor

<sup>1</sup>Though via resistance is not explicitly mentioned in the equation, we add via resistance to the resistance of the interconnect segment which is next to the via.

| Activity Pattern | Input Arrival Time (Agg., Vic.) (ps) | Victim Delay (ps) |              |           |
|------------------|--------------------------------------|-------------------|--------------|-----------|
|                  |                                      | HSpice            | Our Approach | Error (%) |
| ↑↑               | 0, 0                                 | 7.4               | 7.67         | 3.6       |
| ↑↑               | 40, 0                                | 8.65              | 10.12        | 17        |
| ↑↑               | 50, 0                                | 9.73              | 10.73        | 10.3      |
| ↑↑               | 60, 0                                | 10.78             | 11.31        | 4.9       |
| ↑↑               | 0, 40                                | 7.34              | 7.81         | 6.4       |
| ↑↑               | 0, 50                                | 8.17              | 7.84         | 4.0       |
| ↑↑               | 0, 60                                | 9.57              | 8.17         | 14.6      |
| ↑↓               | 0, 0                                 | 14.3              | 15.3         | 7.0       |
| ↑↓               | 40, 0                                | 13.3              | 12.58        | 6.0       |
| ↑↓               | 50, 0                                | 12.2              | 11.9         | 2.5       |
| ↑↓               | 60, 0                                | 10.8              | 11.49        | 6.4       |
| ↑↓               | 0, 40                                | 13.9              | 15.48        | 11.4      |
| ↑↓               | 0, 50                                | 12.7              | 15.52        | 22.2      |
| ↑↓               | 0, 60                                | 11.8              | 13.33        | 13.0      |

Table 1: Comparison of results of our approach versus *HSpice* for a two-line coupled system. All slew times are 100ps.

| Activity Pattern | Input Arrival Time (Agg., Vic., Agg.) (ps) | Victim Delay (ps) |              |           |
|------------------|--|-------------------|--------------|-----------|
|                  |  | HSpice            | Our Approach | Error (%) |
| ↑↑↑              | 100, 100, 100                              | 7.63              | 7.67         | 0.5       |
| ↑↑↓              | 100, 100, 100                              | 14.97             | 15.24        | 1.8       |
| ↓↑↓              | 100, 100, 100                              | 22.46             | 23.83        | 6.1       |
| ↑↑↑              | 50, 100, 50                                | 8.92              | 7.73         | 13.3      |
| ↑↑↓              | 50, 100, 50                                | 14.97             | 15.26        | 1.94      |
| ↓↑↓              | 50, 100, 50                                | 18.63             | 19.25        | 3.3       |
| ↑↑↑              | 150, 100, 150                              | 12.45             | 14.26        | 14.5      |
| ↑↑↓              | 150, 100, 150                              | 14.97             | 15.24        | 1.8       |
| ↓↑↓              | 150, 100, 150                              | 19.08             | 16.26        | 14.8      |

Table 2: Comparison of results of our approach versus *HSpice* for a three-line coupled system. All slew times are 100ps.

and victim arrival and slew times [11] which in turn depend on the switch factors. Therefore, delay and switch factor computation is an iterative procedure. We start with the switch factor of all segments being the same as the first one (which depends on the input arrival and slew times). We then calculate interconnect delays and slew times<sup>2</sup> segment by segment and update the switch factors. We iterate until the sum of delays of all the wires involved changes by less than 5% in two successive iterations.

To test the accuracy of this approach, we compare the results with those of *HSpice* for 1mm long coupled lines in 130nm technology. The parameters used are  $r = 0.098\Omega/\mu\text{m}$ ,  $c_g = 0.0565\text{fF}/\mu\text{m}$ ,  $c_c = 0.078\text{fF}/\mu\text{m}$ , and  $C_L = 50\text{fF}$ . Interconnects are divided into  $100\mu\text{m}$  segments. We simulate two- and three-line systems under a variety of excitation patterns with both aggressor and victim input slew time constant at 100ps. Results are given in Table 2 and Table 1. Our approach gives an average error of 12.2% for the two-line system, and average error of 6.4% for the three-line system. The approach converges in 4-5 iterations; typical runtimes for the three-line system are 0.005s for our approach versus 0.27s for *HSpice* on a 2.4GHz Pentium 4 CPU with 1GB RAM running RedHat Linux 7.3.

## 3 Swizzling

Swizzling is our term for the permutation of long parallel wires. We define the following terminology for swizzling.

- Given a bus of  $n$  wires, swizzling can be done in *swizzle-groups* of  $k$  wires, where  $k$  divides  $n$ . For instance, a 16-wire bus can be swizzled in groups of size 2, 4, 8 or 16.
- A *swizzle-set* is a set of swizzles such that all possible adjacencies within the swizzle-group are realized. For  $k$

<sup>2</sup>Slew time is assumed to be 90% threshold delay and is given by  $2.3T_{ED}$ .

lines, there are  $k(k-1)$  (ordered) pairs of possible adjacent wires. Each wire permutation exhausts  $2(k-1)$  pairs. Thus, the total number of pairwise disjoint permutations is  $k/2$ , implying that the size of a swizzle-set is also equal to  $k/2$ . For example, with  $k=4$ , after 1234, 2413 we cannot construct a third permutation which is pairwise disjoint from the above two permutations. 1234, 2413 thus forms a swizzle-set.

- Any permutation in which wires  $i$  and  $j$  are adjacent is called an  $i$ - $j$  compliant permutation.

### 3.1 Swizzling Objectives

Swizzling targets the routing of long parallel bus-like wires. Possible objectives for swizzling are as follows.

1. *Minimum Delay Uncertainty.* Swizzling displaces aggressors with respect to any designated victim. As a result of this *symmetry breaking*, worst-case coupling does not last for the entire length of the line. Moreover, swizzling reduces the total coupled length of any wire with any other wire in the bus. Limiting the maximum length for which two wires are coupled inserts an element of randomization in the routing. This limits the potential worst-case switching interaction between any pair of wires.
2. *Minimum Layout Overhead.* Swizzling consumes extra routing resources. This overhead increases with the swizzle-group size and the number of swizzles. Therefore, the layout overhead must be traded off against the delay benefit (either in terms of reduction of worst-case delay or reduction of delay uncertainty).

### 3.2 Construction of Swizzling Patterns

Here, we assume without loss of generality the horizontal direction to be the principal routing direction. We now illustrate a “good” swizzling pattern for the case when the swizzle group has size  $k=4$ . Consider 1234, 2413, 4321, 3142 as the swizzling pattern composed of two swizzle-sets 1234, 2413 and 4321, 3142. It has the following desirable properties.

- All possible adjacencies are realized exactly twice. I.e., every wire in the swizzle-group couples to every other wire for the same length.
- The vertical distance traveled by any two wires  $i, j$  between two successive  $i$ - $j$  compliant permutations is the same, and is equal to  $3d$  where  $d$  is the spacing between two nearest-neighbor wires. For example, wires 1 and 2 are adjacent in permutations 1234 and 4321. The vertical distance traveled by each of wires 1 and 2 between these two permutations is exactly  $3d$ .
- The swizzling pattern can be easily routed with a small amount of vias and adjacent-layer routing (including some non-preferred direction routing), without using any extra horizontal tracks on the principal routing layer. Here, the horizontal direction is assumed to be the principal routing direction. The corresponding routing pattern is shown in Figure 1.

The symmetry resulting from these properties ensures that any result derived for any of the wires will hold for the remaining wires as well.

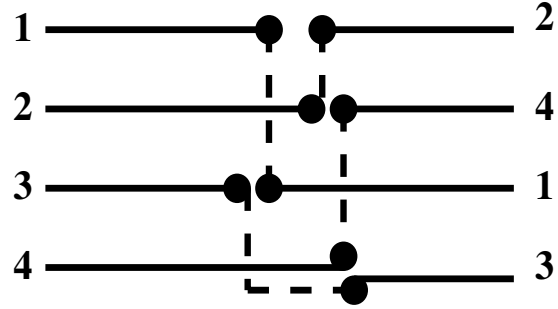


Figure 1: Example routing pattern of 1234, 2413 swizzle-set. Dotted lines denote use of the vertical (orthogonal) routing layer and circles denote vias. Note that wire 3 is routed on the same horizontal track as wire 4, but is drawn a little below 4 for clarity.

More generally, let  $\Pi_i(j)$  denote the wire at the  $j^{\text{th}}$  position in the  $i^{\text{th}}$  permutation in the swizzle-group of size  $k$ . Then, a simple mapping to construct these swizzle-sets for arbitrary even swizzle-group size  $k$  is as follows.

$$\begin{aligned} \Pi_i(k-1) &= \Pi_{i+1}(k) \\ \Pi_i(2) &= \Pi_{i+1}(1) \\ \Pi_i(2r) &= \Pi_{i+1}(2r-2) \quad \forall 2 \leq r \leq \frac{k}{2} \\ \Pi_i(2r+1) &= \Pi_{i+1}(2r+3) \quad \forall 0 \leq r \leq \left(\frac{k}{2} - 2\right) \end{aligned} \quad (3)$$

Moreover, the distance traveled by any pair of wires  $i, j$  between two successive  $i$ - $j$  compliant permutations is exactly  $(k-1)d$ .

A similar swizzling pattern for a swizzle-group size of 6 can be derived as 123456, 241635, 462513, 654321, 536142, 315264. The swizzling pattern so constructed retains the desirable properties mentioned above. The layout overhead per swizzle-set (of  $k$  swizzles) is  $k(k-1)d$  units of vertical routing,  $2k^2$  vias and some wrong direction routing.<sup>3</sup>

### 3.3 Impact of Swizzling on Worst-Case Delay

In this subsection, we analyze the impact of swizzling on delay due to worst-case coupling on a designated victim. Consider a swizzling pattern as described in the previous subsection, with  $k$  swizzles for a swizzle group of size  $k$ . Let wire  $r$  be the designated victim. Now, to impose a worst-case coupling on  $r$ , all other wires must switch in the opposite direction as  $r$ .

Next, for any arbitrary wire  $i \neq r$ , consider two successive  $i$ - $r$  compliant permutations. We note the following.

1. Between the two permutations, wire  $r$  experiences a switch factor between -1 and 3 per aggressor. Wire  $i$  experiences a switch factor between -1 and 1 since all wires in the swizzle group except  $r$  are switching in the same direction as  $i$ .
2. Both wire  $i$  and wire  $r$  are adjacent to any given aggressor exactly once in the  $k/2$  swizzles between the two  $i$ - $r$  compliant permutations.

<sup>3</sup>A minor comment is that since bus routing is performed on semi-global and global routing layers, these vias are being added into relatively wide and long traces. They will therefore tend to be less costly in terms of yield and reliability since they are easy to reinforce in the detailed routing.

- Wire  $i$  experiences “bad” coupling exactly once (when it is a neighbor of  $r$ ) while wire  $r$  never experiences “good” coupling.

From these observations, we may expect that delays and slews along wire  $i$  are smaller than those for wire  $r$ . Therefore, if the arrival (slew) time differences between  $i$  and  $r$  at the beginnings of the two  $i$ - $r$  compliant permutations are, respectively,  $\Delta A_1(\Delta S_1)$  and  $\Delta A_2(\Delta S_2)$ , then  $\Delta A_1 \leq \Delta A_2$  and  $\Delta S_1 \leq \Delta S_2$ . Figure 2(a) depicts a situation when the victim switch factor decreases from 3 to 1. Another case, when the switch factor increases from 1 to 3, is shown in Figure 2(b). A pathological case when the switch factor remains at 3 is shown in Figure 2(c). This case is unlikely to arise as such a large mismatch in slew rates ( $> 2x$ ) is unlikely in practice. Moreover, the next compliant permutation in Figure 2(c) is likely to reduce the waveform overlap between aggressor and victim and hence the victim switch factor. We conclude that the swizzling-induced arrival and slew time displacement implies that worst-case coupling cannot be preserved for any victim along its entire length. However, a formal analysis and proof of this intuition remains an open direction for research.

Here it might be interesting to note that swizzling reduces the chance of the activity pattern which excited worst-case (or best-case) coupling to occur. For instance, assume that all lines in a swizzle-group of size  $k$  switch independently with switching probability of  $A$ . Then the probability of all aggressors (assume only the two nearest neighbors are the aggressors) switching opposite to a designated victim is  $A(\frac{A}{2})^{k-1}$ . In case a swizzling pattern as described in Section 3.2 is used, this is the probability of occurrence of worst-case activity pattern. In the unswizzled case, this probability is  $A(\frac{A}{2})^2$ , i.e., more likely by a factor that is exponential in the size  $k$  of the swizzle-group.

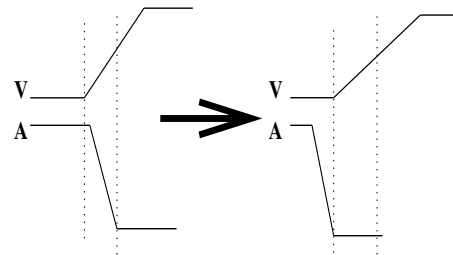
### 3.4 Impact of Swizzling on Best Case Delay

*Minimum* switch factor is experienced by a victim when both of its aggressors are switching in the same direction as itself. In this case, all lines in the bus switch in the same direction and hence displacement in arrival times is minimal. Though swizzling does not affect the switch factors in this case, there is an increase (i.e., worsening) of the best-case delay due to additional interconnect vias inserted due to swizzling.

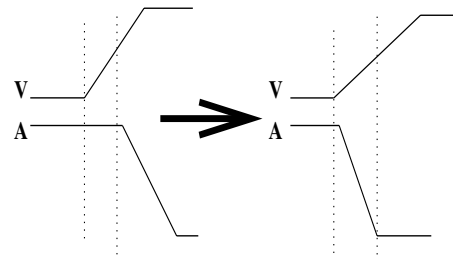
## 4 Experiments and Results

In this section, we experimentally confirm our intuitions regarding the potential performance benefits of swizzling. We describe our experiments and results for delay variation versus number of swizzles, for a set of typical global lines in various technology nodes. We use the delay model described in Section 2 above. To find the worst-case delay, we use simple, iterative greedy search over combinations of aggressor arrival and slew times with respect to a designated victim. We restrict the search space such that no two slew times differ by more than 100% at the beginning of the line. We similarly use an iterative search to identify the best-case delays. Typical runtime for calculation of worst-case or best-case delay is 5 CPU minutes for a swizzle-group size of 4, and 7 CPU minutes for swizzle-group size of 6, using a 2.4GHz Pentium 4 machine with 1GB RAM running RedHat Linux 7.3.

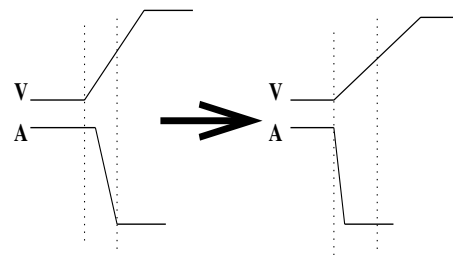
Our studies consider 2mm long global interconnects in 130nm, 90nm and 65nm technologies with swizzle groups of size 4 and 6; the line length reflects typical repeater distances



(a) Switch factor reduction due to swizzling.



(b) Switch factor increase due to swizzling.



(c) Possible no change in switch factor between two  $i$ - $j$  compliant permutations.

Figure 2: Effect of swizzling on victim switch factor between two  $i$ - $j$  compliant permutations.  $V$  represents the victim while  $A$  denotes the aggressor.

| Number of Swizzle-sets | Worst-Case Delay (ps) | Best-Case Delay (ps) | Delay Uncertainty (ps) |
|------------------------|-----------------------|----------------------|------------------------|
| 0                      | 82.4                  | 7.7                  | 74.7                   |
| 1                      | 62.3                  | 11.1                 | 51.2                   |
| 2                      | 67.5                  | 10.3                 | 57.2                   |
| 3                      | 69.9                  | 9.3                  | 60.6                   |
| 4                      | 72.7                  | 9.7                  | 63.0                   |

Table 4: Impact of swizzling on delay uncertainty as calculated by *HSpice* for a 2mm long global interconnect in 130nm technology for a swizzle group of size 4.

in these technologies. Corresponding swizzle-sets are of sizes 2 and 3, and are generated as specified by Equation 3. Interconnect technology parameters are shown in Table 3. The lines in the bus are assumed to be at minimum spacing with a load capacitance of  $50fF$ . Results for various numbers of swizzle-sets are shown in Table 5. For practical reasons, delays have been computed using our approximate delay model. Accurate technology-specific circuit simulations can also be done to suggest design rules for swizzling. As an example, swizzling results with *HSpice* for the 130nm technology node are shown in Table 4. Accurate circuit simulation predicts a 24.4% improvement in worst case delay and 31.5% reduction in delay uncertainty due to swizzling. As expected from the arguments in Section 3, there is a decrease in worst-case delay while there is a slight increase in best-case delay with swizzling.<sup>4</sup> As a result, the delay uncertainty due to capacitive coupling decreases. The observed peak reductions in (worst-case delay, delay uncertainty) for various technology nodes are as follows.<sup>5</sup>

- 130nm: (31.5%, 33.7%)
- 90nm: (25.8%, 32.0%)
- 65nm: (25%, 34%)

The magnitude of these delay and delay uncertainty reductions is noteworthy, and on par with benefits derivable from, e.g., new interconnect materials technology. In modern design methodologies that rely on coupling-aware static noise and timing verification tools, swizzling appears to offer a low-overhead means of reducing guardbands and achieving signoff at higher target frequencies. Of course, actual reductions are design-specific and will depend on specific length, spacing and parasitics of global bus wires. Moreover, since Elmore delay and switch factor based delay estimation both tend to be pessimistic, actual (or, *HSpice*-calculated) delay benefits might be slightly smaller than what we have observed in our studies.

## 5 Conclusions and Future Work

In this work, we have presented *wire swizzling* as a feasible and effective approach to reducing the delay uncertainty worst-case delays that arise from capacitive coupling. We have given good, general swizzling pattern constructions, and empirically computed the optimal number of swizzles for typical global buffered

<sup>4</sup>Worst-case delay is not monotonically decreasing with number of swizzles as the impact of extra wires and vias starts to dominate after a certain optimal number of swizzles.

<sup>5</sup>The reader may point out that this intentional shift in arrival times may be offset to some extent by process variations. However, since swizzling is done on neighboring lines, there would tend to be a strong spatial correlation of variation such that process variations will not tend to impact the relative delays of the swizzled lines.

interconnects in 130nm, 90nm and 65nm technology nodes. Our results indicate the following.

- The maximum reduction in worst-case delays achieved by swizzling are 31.5%, 25.8% and 25% for 130nm, 90nm and 65nm technology nodes respectively.
- The maximum reduction in delay uncertainties achieved by swizzling are 33.7%, 32% and 34% for 130nm, 90nm and 65nm technology nodes respectively.

A large enough delay benefit can lead to reduction in the number of repeaters which can more than compensate for the increase in number of vias due to swizzling. We are currently engaged in further experimental validation of the swizzling approach. For example, we would like to confirm robustness of worst-case delay and delay uncertainty minimization when the locations of swizzles can be perturbed (e.g., due to obstacles or other layout constraints). Another variant arises when one of the wires in an  $n$ -bit bus is critical: the critical wire may need to be swizzled less than others, which disturbs the uniformity of the swizzling pattern. Other ongoing research and future work is in the following directions:

- formal analysis of the worst-case delay impact of swizzling;
- closed-form solution for the optimal number of swizzles

## References

- [1] A. B. Kahng, S. Muddu, E. Sarto and R. Sharma, "Interconnect Tuning Strategies for High Performance ICs", *Proc. IEEE/ACM Design Automation and Testing in Europe*, 1998.
- [2] P. Chen, D. K. Kirkpatrick and K. Keutzer, "Miller Factor for Gate-Level Coupling Delay Calculation", *Proc. IEEE/ACM International Conference on Computer Aided Design*, 2000, pp. 68-74.
- [3] Prashant Saxena and C.L. Liu, "A Postprocessing Algorithm for Crosstalk-Driven Wire Perturbation", *IEEE Transactions on CAD of ICs and Systems*, 19(6), 2000, pp 691-702.
- [4] Kyoung-Son Jhang, Soonhoi Ha and Chu Shik Jhon, "COP: A Crosstalk Optimizer for Gridded Channel Routing", *IEEE Transactions on CAD of ICs and Systems*, 15(4), 1996, pp 424-429.
- [5] Ashok Vittal and Malgorzata Marek-Sadowska, "Crosstalk Reduction for VLSI", *IEEE Transactions on CAD of ICs and Systems*, 16(3), 1997, pp 290-298.
- [6] Desmond A. Kirkpatrick and Alberto L. Sangiovanni-Vincentelli, "Techniques for Crosstalk Avoidance in the Physical Design of High-Performance Digital Systems", *Proc. IEEE/ACM International Conference on Computer Aided Design*, 1994, pp 616-619.
- [7] Desmond A. Kirkpatrick and Alberto L. Sangiovanni-Vincentelli, "Digital Sensitivity: Predicting Signal Interaction using Functional Analysis", *Proc. IEEE/ACM International Conference on Computer Aided Design*, 1996, pp 536-541.
- [8] Tong Gao and C.L. Liu, "Minimum Crosstalk Channel Routing", *Technical Digest Int. Conf. on CAD*, 1993, pp 692-696.
- [9] G. Zhong, C-K Koh and K. Roy, "A Twisted-Bundle Layout Structure for Minimizing Inductive Coupling Noise", *Proc. ICCAD 2000*, pp. 406-411.
- [10] International Technology Roadmap for Semiconductors, 2001, <http://public.itrs.net>
- [11] A. B. Kahng, S. Muddu and E. Sarto, "On Switch Factor Based Analysis of Coupled RC Interconnects", *Proc. ACM/IEEE Design Automation Conference*, 2000, pp. 79-84.
- [12] Star-HSpice, [http://www.synopsys.com/products/avmrg/star\\_hspice\\_ds.html](http://www.synopsys.com/products/avmrg/star_hspice_ds.html)
- [13] XYALIS, <http://www.xyalis.com/> (see data sheets for the GTSmooth product; similar claims of performance-aware fill insertion are made by UbiTech, Inc., <http://www.ubitechnology.com/>).
- [14] K.E. Dow, "Apparatus for improving signal transmission along parallel lines", United States Patent 5,306,967, 1994.
- [15] R. Dutta, Personal communication, 1998.

| Technology Node | Interconnect Width( $\mu\text{m}$ ) | Interconnect Spacing( $\mu\text{m}$ ) | Thickness ( $\mu\text{m}$ ) | ILD Height ( $\mu\text{m}$ ) | $r$ $\Omega/\mu\text{m}$ | $c_g$ $fF/\mu\text{m}$ | $c_c$ $fF/\mu\text{m}$ | Via Res. ( $\Omega$ ) |
|-----------------|-------------------------------------|---------------------------------------|-----------------------------|------------------------------|--------------------------|------------------------|------------------------|-----------------------|
| 130nm           | 0.335                               | 0.335                                 | 0.670                       | 0.670                        | 0.098                    | 0.056                  | 0.078                  | 2                     |
| 90nm            | 0.237                               | 0.237                                 | 0.498                       | 0.498                        | 0.186                    | 0.047                  | 0.072                  | 2                     |
| 65nm            | 0.160                               | 0.160                                 | 0.325                       | 0.325                        | 0.423                    | 0.038                  | 0.055                  | 2                     |

Table 3: Technology parameters for global wires derived from [10].

| Technology Node | Swizzle-Group Size | Number of Swizzle-sets | Worst-Case Delay (ps) | Best-Case Delay(ps) | Delay Uncertainty(ps) | # Vias | Wirelength Overhead ( $\mu\text{m}$ ) |     |
|-----------------|--------------------|------------------------|-----------------------|---------------------|-----------------------|--------|---------------------------------------|-----|
| 130nm           | 4                  | 0                      | 99.2                  | -8.2                | 107.4                 | 0      | 0                                     |     |
|                 |                    | 1                      | 77.9                  | 2.4                 | 75.5                  | 16     | 2                                     |     |
|                 |                    | 2                      | 71.6                  | 0.5                 | 71.1                  | 32     | 4                                     |     |
|                 |                    | 3                      | 73.5                  | 0                   | 73.5                  | 48     | 6                                     |     |
|                 |                    | 4                      | 86.3                  | 2.4                 | 83.9                  | 64     | 8                                     |     |
|                 | 6                  | 0                      | 95.6                  | -8.2                | 103.8                 | 0      | 0                                     |     |
|                 |                    | 1                      | 73.4                  | -3.4                | 76.8                  | 36     | 5                                     |     |
|                 |                    | 2                      | 65.5                  | -3.5                | 69                    | 72     | 10                                    |     |
|                 |                    | 3                      | 70.5                  | -3.4                | 73.9                  | 108    | 15                                    |     |
|                 |                    | 4                      | 79.6                  | -3.5                | 83.1                  | 144    | 20                                    |     |
|                 | 90nm               | 4                      | 0                     | 154.2               | 6.9                   | 147.3  | 0                                     | 0   |
|                 |                    |                        | 1                     | 122.6               | 16.3                  | 106.3  | 16                                    | 1.4 |
| 2               |                    |                        | 114.6                 | 15.2                | 99.4                  | 32     | 2.8                                   |     |
| 3               |                    |                        | 114.4                 | 14.2                | 100.2                 | 48     | 4.2                                   |     |
| 4               |                    |                        | 117.1                 | 14.6                | 102.5                 | 64     | 6.4                                   |     |
| 6               |                    | 0                      | 152.9                 | -0.7                | 153.6                 | 0      | 0                                     |     |
|                 |                    | 1                      | 123.7                 | 6.6                 | 117.1                 | 36     | 3.5                                   |     |
|                 |                    | 2                      | 124.6                 | 11.6                | 113                   | 72     | 7                                     |     |
|                 |                    | 3                      | 121.3                 | 11.2                | 110.1                 | 108    | 10.5                                  |     |
|                 |                    | 4                      | 117.6                 | 11.5                | 106.1                 | 144    | 14                                    |     |
| 65nm            |                    | 4                      | 0                     | 269.1               | 24.2                  | 244.9  | 0                                     | 0   |
|                 |                    |                        | 1                     | 211.4               | 46.1                  | 165.3  | 16                                    | 1   |
|                 | 2                  |                        | 232.6                 | 47.5                | 185.1                 | 32     | 2                                     |     |
|                 | 3                  |                        | 209.6                 | 46.4                | 163.2                 | 48     | 3                                     |     |
|                 | 4                  |                        | 213.3                 | 49.0                | 164.3                 | 64     | 4                                     |     |
|                 | 6                  | 0                      | 283.2                 | 27.1                | 256.1                 | 0      | 0                                     |     |
|                 |                    | 1                      | 213.7                 | 38.1                | 175.6                 | 36     | 2.5                                   |     |
|                 |                    | 2                      | 212.3                 | 43.3                | 169                   | 72     | 5                                     |     |
|                 |                    | 3                      | 217.4                 | 41.4                | 176                   | 108    | 7.5                                   |     |
|                 |                    | 4                      | 216.0                 | 42.4                | 173.6                 | 144    | 10                                    |     |

Table 5: Variation of delay with swizzle-group size, number of swizzle-sets and the technology node. The designated victim is line 2 and its slew time is fixed at 100ps (130nm), 80ps (90nm) or 60ps (65nm). Some negative delay numbers arise from negative switch factors (just as seen in timing reports from commercial STA tools).