COUPLING AWARE ROUTING

Ryan Kastner, Elaheh Bozorgzadeh and Majid Sarrafzadeh

Department of Electrical and Computer Engineering
Northwestern University
kastner,elib,majid@ece.northwestern.edu

ABSTRACT

In this paper, we develop methods to reduce interconnect delay and noise caused by coupling. First, we introduce two novel problems that deal with coupling – the Coupling-Free Routing (CFR) Problem and the Maximum Coupling-Free Layout (MAX-CFL) Problem. We argue that these problems are useful in both global and detailed routing. Then, we develop algorithms to efficiently solve the problems. Our experimental results show that the algorithms work effectively on real data.

1 INTRODUCTION

As fabrication technology moves into deep submicron (DSM) device sizes and gigahertz clock frequencies, interconnect becomes an increasingly dominate factor in performance, power, reliability and cost. In particular, coupling is of greater importance for power, area and timing in circuits. There are four principal reasons for this, increasing interconnect densities, faster clock rates, more aggressive use of high performance circuit families, and scaling threshold voltages.

In order to keep wiring resistance from increasing too quickly, many processes are scaling the wire height at a slow rate (compared to wire pitch). This results in taller, thinner wires. Also, spacing between wires is decreasing in order to yield high packing densities. A detrimental side effect of these scaling trends is an increased amount of coupling capacitance. Coupling capacitance is proportional to the amount of parallel overlap between the wires and inversely proportional to the distance between the wires (coupling is formally define in Section 2). In fact, coupling capacitance between wires can account for over 70% of the total wiring capacitance, even in 0.25 μm processes [2]. There are two problems introduced by coupling, delay deterioration and crosstalk [8].

In this paper, we focus on reducing the unwanted effects caused by coupling. In Section 2, we give some basic definitions, formally define coupling and introduce the Coupling-Free Routing (CFR) Problem. Section 3 introduces the Maximum Coupling-Free Layout (MAX-CFL) Problem and analyzes two algorithms developed to solve the problem. We conclude in Section 4.

2 PRELIMINARIES

A *multi-terminal net* n $= \{(x_1, y_1,), (x_2, y_2), (x_3, y_3), ..., (x_n, y_n)\}$ is a collection of points in the plane. A *terminal* is single point of a net. A multi-terminal net can be partitioned into a collection of *two-terminal* nets (a net with exactly two points) using a number of standard techniques.

A two-terminal net (or simply called a *net* hereafter) $n = \{(x_1, y_1), (x_2, y_2)\}$ is an unordered pair of points (x_1, y_1) and (x_2, y_2). A *routing* or *wiring* of n is a set of horizontal and vertical line segments connecting (x_1, y_1) and (x_2, y_2). A *layout* is the routings of a set of nets.

A net n can be routed without any bends if and only if either $x_1 = x_2$ or $y_1 = y_2$. We call such a net a *zero-bend net*. Otherwise, there are two ways to route n with one bend as shown in Figure 1. When a routing has no more than one bend, it is called a *single bend routing*.

The routings in Figure 1 are called the *upper-L routing* and the *lower-L routing*. To avoid confusion, we often refer to a possible routing as a *route*. Thus we say that a one-bend net has two one-bend routes (the upper-L route and the lower-L route).

We focus on routing *critical nets*. Critical nets can be defined in a variety of ways. Most often, critical nets correspond to nets that are on a critical path of a network at the logic synthesis stage. Interconnect delay of the these nets should be minimal, therefore we are interested in a single bend routing. A single bend routing is not only the shortest possible route

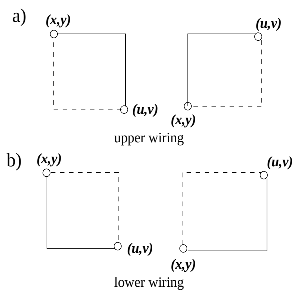

Figure 1: **a)** Upper-L routings **b)** Lower-L routings

between the two terminals, it also introduces only one via. Since vias further increase the wire capacitance and resistance, it is beneficial to keep them at a minimum. Also, vias negatively effect the routability of the circuit.

A) Coupling

The coupling capacitance C_C between two wires i and j can be represented as follows:

$$C_C(i,j) = \frac{f_{ij} \cdot l_{ij}}{d_{ij}} \frac{1}{1 - \frac{w_i + w_j}{2d_{ij}}} \quad (1)$$

where w_i and w_j are the sizes of wires i and j ($w_i, w_j > 0$), f_{ij} is the unit length fringing capacitance between wires i and j, l_{ij} is the overlap length of wires i and j and d_{ij} is the distance from the center line of wire i to the center of wire j (see Figure 2).

We are trying to minimize the coupling. During routing, we can control l_{ij}, d_{ij}, w_i and w_j. By avoiding overlap between two wires, l_{ij} can be minimized. In other words, we do not want adjacent wires to run in parallel for long distances. We assume that w_i, w_j, l_{ij} are fixed; we do not consider wire sizing and spacing in our algorithm. But, this can easily be done as a post-processing step.

B) Coupling-Free Routing

Every wire consists of horizontal and/or vertical line segments. We say two wires *couple* if the line segments forming them are closer than d units for more than l units. Two line segments intersect if they have at least one point in common and overlap if they have more than one point in common.

For a given set of n nets $S = \{n_i = \{(x_{1i}, y_{1i}), (x_{2i}, y_{2i})\} \mid 1 \leq i \leq n\}$, a (single bend) layout of S is a coupling-free routing if there are no two routes that run in parallel at a distance equal to or closer than d units for more than l continuous units. Examples of coupled and non-coupled layouts

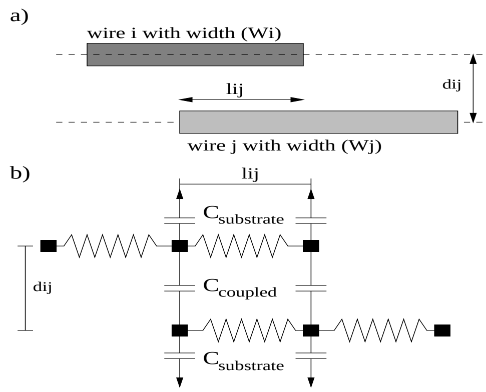

Figure 2: **a)** Physical coupling capacitance between two wires **b)** The wires modeled by resistors and capacitors

are given in Figure 3. Given a set of two-terminal nets, the problem of obtaining a coupling-free routing of nets is called the *coupling-free routing problem* (CFR problem).

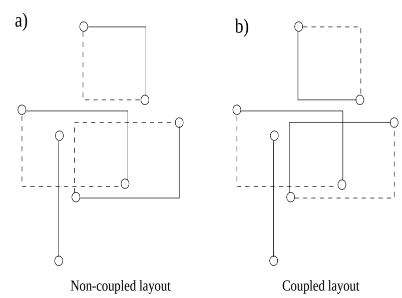

Figure 3: **a)** Coupling-free routings **b)** Non-coupling-free routings

We feel that CFR is beneficial in both detailed and global routing:

As VLSI fabrication technology progresses, more routing layers become available. Therefore, we can afford to set aside *preferred layers* for critical nets. A preferred layer usually has a lower wiring resistance due to position of the layer (lower layers have lower resistance) and width of the wires on that layer (large wire widths have lower resistance). Power, ground and clock nets are already routed on preferred layers. We propose using the preferred layers for routing critical nets. Critical nets are allotted very little slack in order to meet timing constraints. Since interconnect is becoming a dominate factor in delay of a circuit and coupling plays are large role in interconnect delay, these nets should routed in order to minimize coupling and wirelength. Therefore, we can use notion of coupling-free routing to provide a detailed routing for the critical nets. Since the nets are routed with at most one bend, they have minimum wirelength. In addition, coupling-free routing minimizes the coupling of the routed nets. Combining these two factors, we have a routing of the critical nets with minimal interconnect delay. After we have a coupling-free layout, non-critical nets can be routed on the preferred layers to maximize routing resources.

Many single-layer routing algorithms have been suggested. Liao *et. al* [6] propose density routing or maze routing to perform this task. A more recent paper by Lin and Ro [10] improves on the work by Liao *et. al.* They employ a two step process. First, they find a planar set of single-bend nets. Then, they use a method based on rubber-band equivalent to find a routing for the remaining nets. CFR can easily be incorporated into the first stage of Lin and Ro's algorithm to obtain a planar layout that is coupling-free.

Generally, coupling at the global routing stage is hard to determine. A global route is not exact. Therefore, a net could possibly couple with every net that is routed in the same global bin. But, the net will only couple with it's two neighbors[1]. Ultimately, track assignment (which can be done at the global or detailed routing stage) determines the coupling. Additionally, the detailed router will often make local changes which can effect the coupling of nets [4]. But, the detailed router can only make local changes, therefore considering coupling at the global stage, even if it isn't exact, is beneficial as it can provide a way to make large scale changes to a layout that otherwise can not be done at the detailed level. If we have coupling-free layout at the global stage, then the layout will remain coupling-free at the detailed stage. Therefore, we can use CFR at the global routing stage to minimize coupling for the detailed router.

In the next section, we describe a couple heuristics for solving the *maximum coupling-free layout problem* – the maximum number of nets that can be laid out in a coupling-free fashion.

3 MAXIMUM COUPLING-FREE LAYOUT

The Maximum Coupling-Free Layout Problem (MAX-CFL): Given a set of two-terminal nets S and a positive integer $K \leq |S|$. Is there a single-bend routing for at least K nets in S such that no two routings couple?

[1]Theoretically, a net couples with every net on the chip. But, the neighboring nets will act as a shield which makes the coupling capacitance seen by the other nets minimal.

Theorem 4.1: The Maximum Coupling-Free Layout Problem for planar layouts is NP-Complete. [8]

Since MAX-CFL is NP-Complete, we look at heuristic algorithms to solve the problem.

A) Greedy Algorithm

The first and most obvious algorithm that we consider is the greedy algorithm. This algorithm chooses the most critical net and, if possible, routes the net in an upper-L or lower-L fashion. If both the upper-L and lower-L routings couple with net that has already been laid out, the current net is not laid out and the most critical remaining net is considered. The algorithm iterates until all nets have been considered.

Algorithm 1 *Maximum Coupling-Free Layout Routing Greedy Heuristic*

```
Given a set of nets N
Sort N by criticality
for(i ← 0 to |N|)
    route N[i] in upper-L or lower-L, if possible
```

Theorem 4.2: The Maximum Coupling-Free Routing Greedy Heuristic takes $O(n \log n)$ time [8].

Therefore, the greedy heuristic is a simple and fast method of finding a maximum coupling-free layout solution.

Of course, there are many shortcomings to this algorithm. First, the greedy nature of the algorithm may cause a critical net that couples with many other less critical nets to be routed. By not routing a critical net, you may be able to route a large number of other less-critical nets which can lead to a better overall solution.

B) Forcing Algorithm

A routing of a net may *force* a routing of another net. For example, assume net A is routed in an upper-L. If the upper-L routing of A couples with the lower-L routing of B, then net B must be routed as an upper-L to avoid coupling. Hence net A forces net B.

The forcing algorithm tries to eliminate the the bad decisions made by greedy algorithm. It starts by determining the forcing interactions between every pair of nets. Then, it finds any nets that have a truly independent routing (either upper-L or lower-L) and routes them in the appropriate manner. An independent routing is equivalent to a route that forces no other nets If a net only forces other nets when it is routed in a lower-L (upper-L) will be routed in an upper-L (lower-L). The remaining nets are routed according to the number of nets that they force. The net that forces the least amount of other nets is routed first, as long it doesn't couple with any net that is already routed. This process continues until all of the nets have been considered.

Algorithm 2 *Maximum Coupling-Free Layout Routing Forcing Heuristic*

```
Given a set of nets N
Determine the forcing interactions between then nets N
Route any independent nets
N' = N− independent nets
R ← ∅
for(i ← 0 to |N'|)
    R = R ∪ N'[i].upper-L ∪ N'[i].lower-L
Sort R by number of forcings
for(i ← 0 to |R|)
    if net associated with R[i] is unrouted and R[i] is
    routable
        route R[i]
```

Theorem 4.3: The Maximum Coupling-Free Routing Forcing Heuristic takes $O(n^2)$ time [8].

C) Evaluation

To perform our experiments, we used five MCNC standard-cell benchmark circuits [5] and five benchmarks from the ISPD98 suite [1]. The characteristics of the circuits are shown in Table 1. The MCNC circuits were placed into using the Dragon global and detailed placement engine [7] which is comparable in quality to commercial version of Timberwolf [9]. The ISPD98 benchmarks are slightly modified so that they could be placed by the Timberwolf placement engine.

Our experiments focus on reducing the added delay caused by coupling. Therefore, we look at the large nets from each of these circuits. We assume that the large nets will be placed on a single preferred layer.

We compare the greedy algorithm and the forced algorithm in terms of number of nets routed and criticality of the nets that are routed. Net criticality is normally defined at the logic synthesis stage and is a function of the amount of slack available on a net. Unfortunately, the Dragon placement engine does not include timing information. Hence, we need another measure of criticality. It has been shown that the delay for a wire of length l increases at the rate of $O(l^2)$ without wiresizing, $O(l\sqrt{l})$ with optimal wiresizing and linearly with proper buffer insertion [3]. We did experiments using both the linear (l) and quadratic (l^2) functions. The criticality function can easily be changed to incorporate wiresizing and buffering.

Data file	Num Cells	Num Nets	Num Pins
MCNC benchmarks			
prim1	833	1156	3303
prim2	3014	3671	12014
avqs	21584	30038	84081
biomed	6417	7052	22253
struct	1888	1920	5407
ISPD98 benchmarks			
ibm01	12036	13056	45815
ibm05	28146	29647	127509
ibm10	68685	75940	298311
ibm15	161187	186991	716206
ibm18	210341	202192	819969

Table 1: Benchmark circuit information

Figure 4 shows the fraction of nets that are placed by both the greedy and forced algorithm. The x-axis corresponds to the number of total nets considered. We can see that the forced algorithm consistently finds a layout for a larger percentage of nets. Over all the experiments that we ran, the forced algorithm routes, on average, 3.38% more nets than the greedy algorithm.

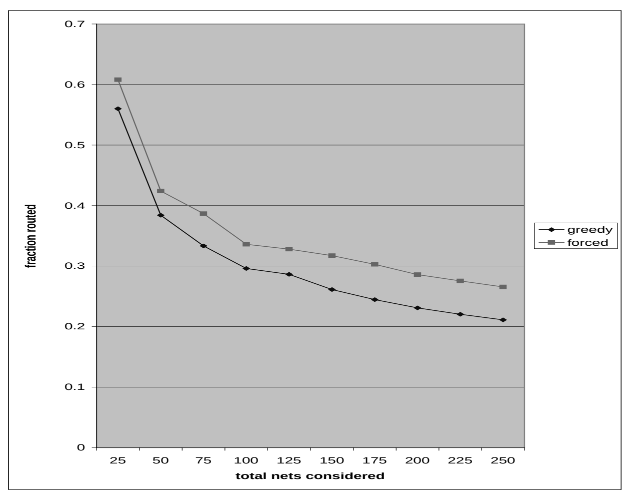

Figure 4: Fraction of nets placed averaged over all benchmarks.

If we only look at the criticality of the nets routed, we see that the greedy algorithm is better than the forced algorithm. Figure 5 confirms that the greedy algorithm outperforms the forced algorithm using a quadratic function and a linear function. For a linear criticality function, the greedy algorithm was approximately a factor of 1.2 times better than the forced algorithm. If we use the quadratic function, the greedy function outperforms the forced function by a factor of 2.54 (when we consider the 250 most critical nets). This should be of little surprise, however, since the forced algorithm does not use the idea criticality to find a routing of the nets.

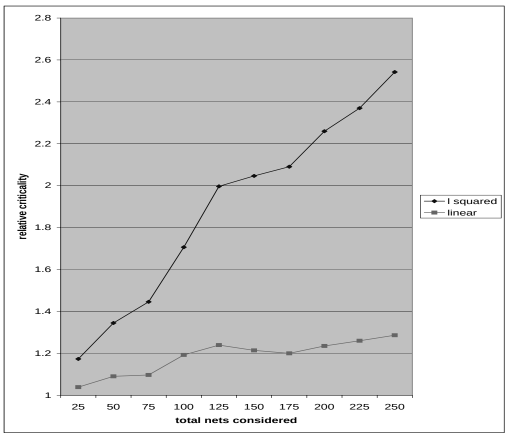

Figure 5: Relative criticality of nets placed by the greedy algorithm compared to the forced algorithm. The results are averaged over all benchmarks. The criticality of the benchmarks are normalized to the criticality result of the forced algorithm. Therefore, a result of x indicates that the greedy algorithm laid out $x \times$(criticality of forcing algorithm).

4 CONCLUSION

In this paper, we present the Coupling-Free Routing (CFR) Problem and the Maximum Couping-Free Layout (MAX-CFL) Problem. We argue that these problems are useful in detailed routing since they:

1. can insure that critical nets are routed with minimum delay on a preferred layer.

2. can be incorporated into existing single layer routings algorithms, making these algorithms more useful as we go further into the DSM era.

Additionally, these problems are useful in global routing to help guide the global router to a coupling-free detailed layout.

We present two heuristics for solving the MAX-CFL problem. Using experimental data, we showed that the greedy algorithm is a simple, but effective way of obtaining a layout with maximal criticality. The forced algorithm gives a layout with the maximal number of nets.

In the future, we plan on developing better algorithms for solving the MAX-CFL problem. Specifically, we hope to incorporate criticality data into the forced algorithm to improve its performance with respect to criticality. Also, we plan on incorporating these algorithms into our existing global and detailed routers.

References

[1] C. Alpert. "The ISPD98 Circuit Benchmark Suite". In *Proc. International Symposium on Physical Design*, April 1998.

[2] D. Sylvester *et al.* "Interconnect Scaling: Signal Integrity and Performance in Future High-speed CMOS Designs". In *Proc. of VLSI Symposium on Technology*, 1998.

[3] J. Cong and D.Z. Pan. "Interconnect Delay Estimation Models for Synthesis and Design Planning". In *Proc. Asia and South Pacific Design Automation Conference*, January 1999.

[4] J. Cong, J. Fang and K.-Y. Khoo. "DUNE: A Multi-Layer Gridless Routing System with Wire Planning". In *Proc. International Symposium on Physical Design*, April 2000.

[5] K. Kozminski. "Benchmarks for Layout Synthesis - Evolution and Current Status". In *Proc. ACM/IEEE Design Automation Conference*, June 1991.

[6] K.F. Liao, M. Sarrafzadeh and C.K. Wong. "Single-Layer Global Routing". In *IEEE Transactions on Computer Aided Design*, 1994.

[7] M. Wang, X. Yang and M. Sarrafzadeh. "DRAGON: Fast Standard-Cell Placement for Large Circuits". In *Proc. IEEE International Conference on Computer Aided Design*, November 2000.

[8] R. Kastner, E. Bozorgzadeh, and M. Sarrafzadeh. "Coupling Aware Routing". In *Technical Report, Northwestern University*, August 2000.

[9] W.J. Sun and C. Sechen. "Efficient and Effective Placement for Very Large Circuits". In *Proc. International Conference on Computer Aided Design*, November 1993.

[10] Z.-M. Lin and Z.-W. Ro. *"A Heuristic Planar Routing Algorithm for High Performance Single-Layer Layout"*. Manuscript, 2000.