

Crosstalk Reduction in Area Routing

Ryon M. Smey
Binghamton University
Binghamton, NY

Bill Swartz
InternetCAD Inc
Dallas, TX

Patrick H. Madden
Binghamton University
Binghamton, NY

Abstract

Interconnect delay dominates system delay in modern circuits, and with reduced feature sizes, coupling capacitance and signal crosstalk have become significant issues. By spacing interconnect wires effectively, and avoiding long parallel runs, coupling can be reduced; with current routing methods, however, this is difficult to achieve.

In this paper, we present a new approach to area routing. Rather than inserting routes sequentially (using a performance driven maze router), multiple candidate routes for each connection are generated; excess routes are then eliminated iteratively. Experiments show that we obtain substantial reductions in coupling capacitance, without sacrificing routing completion rates.

1 Introduction

Interconnect delay now dominates system delay in many circuits, and the coupling capacitance between adjacent wires can significantly impact performance. Traditional area routing methods based on maze routing have difficulty in addressing this issue; in this paper, we present a routing approach that can optimize coupling capacitance, and that complements traditional methods. The work presented has been a collaborative effort between academic and industry researchers; we focus on the academic routing tool here.

In the remainder of this paper, we first discuss the area routing problem, and briefly consider related earlier work. We also summarize physical constraints which have been a motivating factor in our work. We next outline our approach to this problem, illustrated with small examples. Our experimental results focus on routing completion rates (which reflect achievable routing density) and capacitance estimates for problems that are intentionally made difficult; we also show portions of routings from industrial circuits. We conclude this paper with a summary of results, and suggestions for future work.

2 Previous Work and Objectives

Area routing is the most difficult and general form of detail routing. Informally, detail routing problems with connections on two sides of a rectilinear region are *channel routing* problems (for example [7, 11, 13]). If there are connections on four sides of a region, we have a *switchbox* routing problem[12, 3]. If there are also connection points within the routing region, we have an *area routing* problem[13].

While channel routing is essentially solved, switchbox and area routing are still quite difficult. The central problem which makes both switchbox and area routing difficult is that one route may block another, preventing a solution from being found. Most area routers utilize shortest path algorithms to connect pairs of points; while finding a shortest path can be done in polynomial time, finding routes for all connections is NP-complete.

When routing cannot be completed easily, heuristic methods are employed. The most common method, **rip-up and reroute**, is relatively simple. When routes cannot be completed, some connections that have been made successfully are removed, and the routes that were previously blocked are routed in the space that was made available. This approach is widely used; it is time consuming and does not guarantee a solution.

Our routing formulation assumes that the circuit routing has been decomposed into a set of “tiles” or “global routing cells”, as is done in [15][4][9]. Each tile contains a number of pins which must be connected; we decompose multi-pin nets using an efficient Steiner tree heuristic[2]. Our objective is to find precise routings within each tile for each signal net, obeying design rules, and with consideration of performance constraints. There may be specific restrictions on the routing of a connection: a net may be noise-sensitive, require unusual width or layer assignment, and may possibly require shielding. We wish to avoid if possible the computationally expensive rip-up and reroute process. With area routing problems, there are essentially three issues to consider.

- The first problem is finding a path between a pair

points. There is an abundance of research on this problem, starting with maze search on graphs[6][10]. Much of the work on area routing has focused on improving the speed of maze routing, and adding consideration of physical constraints[8, 5]. While finding a path between a pair of points can be time consuming, and efficient implementation of the algorithms is difficult, there are satisfactory solutions to this problem.

- Second, physical and performance constraints must be considered. We summarize these issues as follows. **Delay:** increased routing lengths generally increase delay; to meet performance constraints, a routing solution may have restricted lengths for some connections. **Coupling Capacitance:** in general, two routing lines can act as a capacitor. If the distance between a pair of lines is small, and they run in parallel for a significant distance, the capacitance introduced can degrade performance substantially. In some cases, it is possible that the coupling between a pair of routes will result in a spurious transition. **Design Rules Correctness:** it is common to have spacing requirements for vias and lines that differ on a layer-by-layer basis, and the allowed locations for some features may depend on the other nearby features. Simple maze-routing methods can fail to address these issues correctly, and developing a design-rule correct maze router is nontrivial.
- The third and most important issue to consider in area routing is ensuring that all required paths can be made. As was mentioned, traditional methods utilize extensive rip-up and reroute. We are interested in an alternate method, which will provide better solution quality with lower run times.

There are some similarities in our approach to prior work on satisfiability-based routing (for example [14]). Segments that may be used to complete a routing are mapped to boolean variables, and a boolean equation can be constructed to model the design rule constraints. There are also some similarities in our approach to a recent multi-commodity flow based global router[1], which uses randomized rounding to select from a set of Steiner trees for each net.

3 Our Approach

Overview of Approach

To complete a set of connections, we first generate *multiple* possible connections between points, using a simple combinatorial approach. If we wish to connect pin A to pin A' , we might consider single-bend routes on a variety of layers; these can be enumerated easily. We refer to each route as a *candidate route*, and the set of routes connecting A to A'

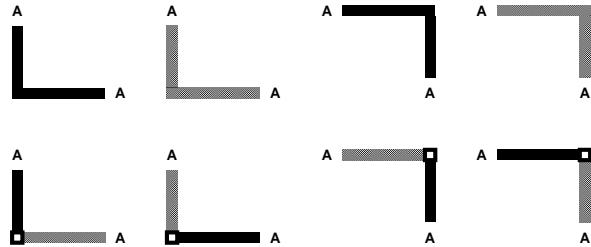


Figure 1. Candidate single-bend routes for two points, using two metal layers.

as a *bundle*. A pair of points, and a set of candidate routes, is shown in Figure 1. Each candidate route is evaluated only on its functionality: it must connect a pair of points, but we place no constraints on the wire sizes (other than minimum width), number of vias, number of bends, or layers used. A routing solution will contain only a single route for each connection; if there is more than one route for a connection, we have *excess* routes. Each generated route is design rule correct by construction, but a pair of routes may violate design rules or even intersect.

We model the interactions between candidate routes with a *constraint graph*: in this graph, each candidate route is represented by a vertex, and design rule violations between routes are represented by edges. This is perhaps the most dramatic departure from previous methods.

A small routing problem is illustrated in Figure 2(a). In this figure, we have three pairs of points which must be connected; each pair of points has two candidate routes. The constraint graph is shown below, modeling the design rule violations between pairs of routes. Determination of the routing solution is through the *elimination* of excess routes; we repeatedly select a (redundant) route which has high cost, and remove it.

Candidate Route Generation

If we have l routing layers, and a pair of points to connect at locations (x_1, y_1) and (x_2, y_2) , we can enumerate L-shaped and Z-shaped routes using various layers. Routes may be restricted to preferred direction if desired; we find that when non-preferred direction routing is used, subsections of the problem naturally organize themselves as preferred direction. The tool developed by the academic group can either generate routes using extremely simple rules or routes that comply with MOSIS design rules.

Route Elimination

After route generation, we must eliminate excess candidate routes from each connection, until only a single route remains for each connection. If two candidate routes from different bundles intersect (or violate design rules), we say that the bundles *interact*. The *cost* of an individual route is

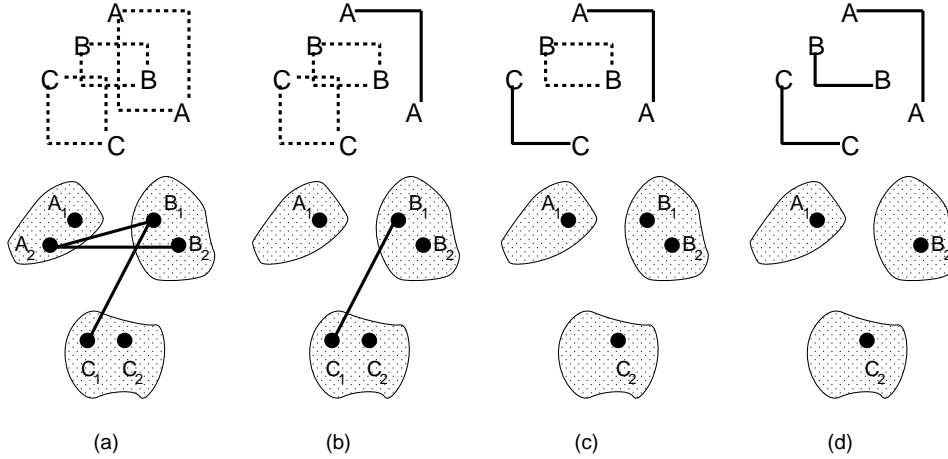


Figure 2. Our routing approach first generates a number of candidate routes for each net, and then removes excess routes to eliminate design rule violations (or to minimize coupling capacitance on critical-path connections).

a function of the route length, and the number and nature of interactions with other candidate routes. The *cost* of a bundle is a function of the costs of each candidate route within the bundle.

The route elimination process operates by selecting the highest cost (excess) route, and then eliminating it. Costs for remaining routes are recalculated, and the process continues as long as excess routes remain.

Route Costs

Each candidate route has a cost based on the length of the route (and the number of vias required to realize it), and also on the number and type of interactions with other candidate routes.

If a candidate route r_i intersects with route r_j from another bundle, we add a penalty cost $p_{intersect}$ to the cost of r_i . If a candidate route r_i violates design rule spacing with route r_j , we add a penalty cost $p_{spacing}$. If a candidate route r_i has a coupling capacitance c with route r_j , we add a penalty cost $c \times p_{coupling}$.

In practice, we set $p_{intersect}$ to a large value (1000), $p_{spacing}$ (900), to a slightly smaller value, and $p_{coupling}$ to a small value (1). This weighting results in the removal of routes that intersect or violate design rules, with the impact of coupling capacitance acting as a “tie-breaker.”

Coupling Capacitance Optimization

In practice, we find that there are many instances where there are several possible ways to complete a connection. By making a preference for routes that are not adjacent, we can reduce coupling capacitance; experiments described below show that this can be surprisingly effective. By contrast, consider the behavior of maze-routing based methods. If Dijkstra’s algorithm is used, a route will “turn” when it

encounters an obstacle; this can result in the route running parallel to an existing feature, with the two routes having significant coupling.

Summary of Approach

The routing approach considers *all* required connections at the start of processing, and eliminates routes that cause the greatest difficulty. At the very earliest stages of detail routing, we can easily determine both *best case* and *worst case* conditions for each candidate route. This allows the determination of *best case* and *worst case* system level performance, with these bounds becoming progressively tighter as the routing process progresses.

The *best case* performance of a route is one in which all other nearby candidate routes have been eliminated, removing coupling capacitance. The *worst case* performance would have the route with highest coupling capacitance from each nearby bundle contributing to the delay. The best case and worst case performance of each bundle can be determined from their candidate routes. This allows determination of system level best case and worst case performance, resulting in our ability to detect problems at an early stage, and to optimize the circuit in the most critical areas first.

4 Experimental Results

In this section, we show experimental results obtained by our new routing tools. Routing results from the academic routing tool are shown in Figures 3(a) and (b). These figures show routes that use simplified design rules. The commercial routing tool was tested on heavily congested portions

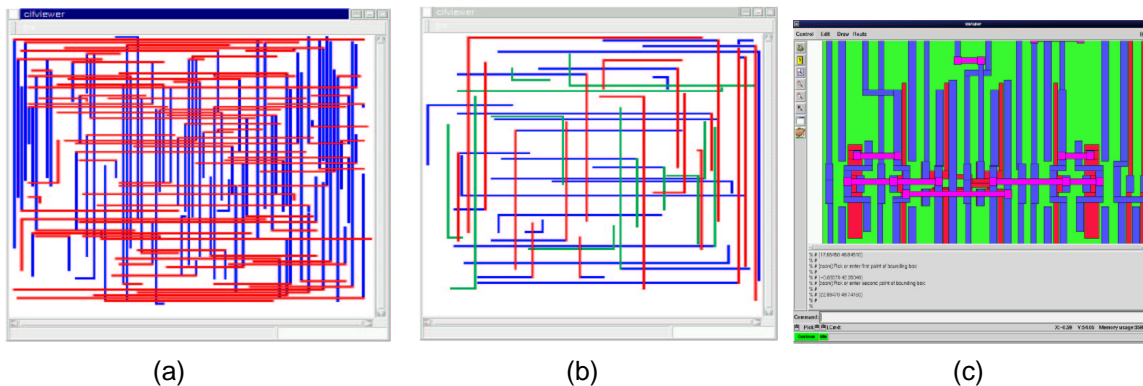


Figure 3. Three routing problems completed by our tools. The first two problems are routed with the academic tool, while the third is an industrial design routed by the commercial tool.

of industry designs; a portion of one routing is shown in Figure 3(c).

The experiments we report here use the academic routing tool, and consider the new approach, a traditional maze-routing based approach, and our new approach followed by maze routing. We also show results when we consider coupling capacitance as part of our optimization objective.

4.1 Comparison with Maze Routing

To allow direct comparison with traditional methods, we first show net completion rates of our new approach and a maze routing based approach on a number of randomly generated problems. We show results for four layer routing problems in Figure 4. In these experiments, pairs of points are randomly distributed on a 200 by 200 grid, with each routing problem having from 10 to 150 pairs. We report the *average* number of routes completed by our new approach on 10 different problems. This Figure shows the following points.

- Route completion rates for our method and traditional maze routing are comparable. While the type of routes considered by our method are restricted to L and Z shapes, this does not dramatically reduce the number of routes we can complete.
- A hybrid routing method obtains the best results; we first find many routes using our method, and then insert additional routes with a maze router. The maze router is able to find connections that require multiple bends, and these cannot be generated by our enumeration process. Maze routing alone has a lower completion rate, however: because the routes can have multiple bends, early routes can create obstacles that later routes have difficulty in avoiding.

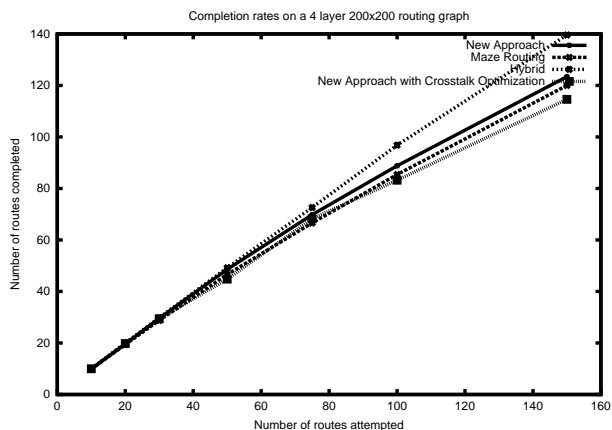


Figure 4. Completion rates for a traditional maze-routing approach, our new approach, and our new approach followed by post-processing by maze routing.

- We include completion rates in this Figure where we optimize coupling capacitance; the achievable density is close to maze routing and our approach.

In other experiments using one to three layers, and on different grid sizes, our results are similar.

4.2 Coupling Capacitance Optimization

Coupling capacitance, as well as other physical effects, depend on wire adjacencies, and this is extremely difficult to control with maze-routing approaches. This issue was a motivating factor in our work, and effectively addressing these interactions was one of our objectives.

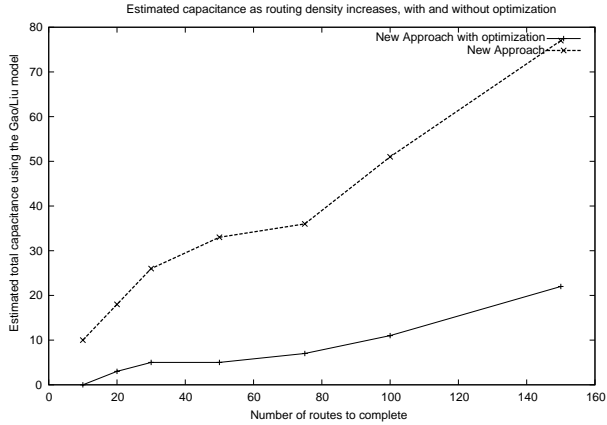


Figure 5. When we consider coupling capacitance in our optimization, we can obtain routes that significantly reduce this effect, simply by better distribution of routes. As routing density increases, so does coupling capacitance.

In our next set of experiments, we show the *estimated coupling capacitance* of the interconnect wires. The academic routing tool uses the simplified model of Gao and Liu[7]; routes which are adjacent incur “one unit” of coupling capacitance penalty per unit length. By incorporating this objective into our cost function, we obtain dramatic decreases in total coupling capacitance (and also significantly lower peak capacitance). These experiments are illustrated in Figure 5. We have again a 200 by 200 graph, 4 layers, and 10 to 150 pairs of pins. We report the *total capacitance* along all connections, and compare to the results of our tool *without* consideration of this effect. Total completion rates of the two methods are similar, and were included in Figure 4. The main points we wish to stress are the following.

- By simply avoiding placing routes adjacent to one another, we can reduce coupling capacitance significantly. There can be many ways to route a set of connections, and some routings are better than others.
- Coupling capacitance optimization does not necessarily mean poor routability. Figure 4 shows that completion rates are comparable.

4.3 Algorithm Run Times

As our new routing approach considers the connections from a geometric perspective, and does not use a grid-based model, run times are independent of grid size. In experiments with small problems, the run times for maze routing

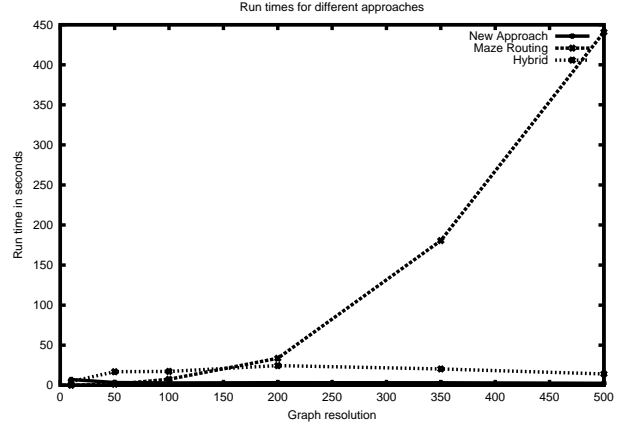


Figure 6. Run times (in seconds) for the routing tools. With large graphs, the maze routing approach incurs a substantial increase in run time.

and our approach were comparable. For larger graphs, the $O(n \log n)$ performance of our approach was substantially better than quadratic (in terms of graph resolution) performance observed with the maze-routing approach. Run times for problems with 100 pairs of points on a variety of graph sizes are shown in Figure 6. The main points we wish to stress are the following.

- Our new approach is extremely fast, and independent of the size of an underlying routing grid. Because the constraint graph is generated using computational geometry methods, run time is only affected by the number of routes that need to be completed; variable width routing, tapered wires, and other issues that are extremely difficult to handle with maze routing do not impact our method at all.
- A hybrid approach is substantially faster than the maze routing approach. Most routes are completed quickly with our new method, leaving only a few routes to be handled by the more computationally expensive maze router. Total run time is thereby substantially reduced.

5 Summary and Conclusion

Area routing will become progressively more difficult, as the constraints on interconnect wires become more demanding. Variable width wiring is supported on only some area routing tools, and implementation of crosstalk optimization is even more difficult. As feature sizes scale down, there will be even greater demand for tools that support novel interconnect structures, and there may eventually be a need

for direct consideration of lithographic issues *during* physical design. Maze-routing based approaches are pushed to their limit; we expect that future design tools will need to utilize alternate methods.

In this paper, we have presented an unusual approach to area routing; this has been implemented in two routing tools, one developed by an academic group, and a second developed by a commercial CAD tool developer. We find that a hybrid, using our new approach for routing the majority of connections quickly, followed by more computationally intensive maze routing, produces the best results.

We have also shown that our approach can handle physical constraints effectively; we have integrated a simple coupling capacitance model into the academic tool; this has resulted in substantial reductions in estimated capacitance. We are currently developing a more accurate capacitance estimation method.

Despite the simplicity of the approach, we can obtain *higher* completion rates than maze-routing, which has been the workhorse of traditional area routing tools. We can also easily and directly support variable width and spacing of wires. Complex design rules, which will become progressively more demanding as device sizes scale down, can be handled easily. The method is computationally efficient, and we do not need to restrict routes to preferred routing directions or specific layers to improve run times or route completion rates.

Our current work involves integration of this approach with global routing, and adding support for performance driven interconnect structures. We are also working with a lithography group to explore methods that will improve manufacturability while reducing mask cost.

Acknowledgement: this work was supported in part by the National Science Foundation under grant CCR-9988222.

References

- [1] C. Albrecht. Global routing by new approximation algorithms for multicommodity flow. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 20(5):622–631, May 2001.
- [2] M. Borah, R. M. Owens, and M. J. Irwin. An edge-based heuristic for Steiner routing. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 13(12):1563–1568, December 1994.
- [3] J. P. Cohoon and P. L. Heck. Beaver: a computational geometry based tool for switchbox routing. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 7:684–697, 1988.
- [4] J. Cong and P. H. Madden. Performance driven multi-layer general area routing for pcb/mcm designs. In *Proc. Design Automation Conf*, pages 356–361, 1998.
- [5] Jason Cong, Jie Fang, and Kei-Yong Khoo. Via design rule consideration in multilayer maze routing algorithms. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 19(2):215–223, 2000.
- [6] E. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.
- [7] T. Gao and C. L. Liu. Minimum crosstalk channel routing. In *Proc. Int. Conf. on Computer Aided Design*, pages 692–696, November 1993.
- [8] S.-W. Hur, A. Jagannathan, and J. Lillis. Timing driven maze routing. In *Proc. Int. Symp. on Physical Design*, pages 208–213, 1999.
- [9] Ryan Kastner, Elaheh Bozozgadeh, and Majid Sarrafzadeh. Predictable routing. *Proc. Int. Conf. on Computer Aided Design*, pages 110–113, 2000.
- [10] C. Y. Lee. An algorithm for path connection and its applications. *IRE Transactions on Electronic Computers*, EC-10(3):346–365, 1961.
- [11] Xiaolin Liu and Ioannis G. Tollis. Improving over-the-cell channel routing in standard cell design. In *Proc. Int. Conf. on Computer Aided Design*, pages 606–609, 1994.
- [12] H. Shin and A. Sangiovanni-Vincentelli. Mighty: a rip-up and reroute detailed router. In *Proc. Int. Conf. on Computer Aided Design*, pages 2–5, 1986.
- [13] Hsiao-Ping Tseng and Carl Sechen. A gridless multi-layer router for standard cell circuits using CTM cells. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 18(10):1462–1479, 1999.
- [14] R. Glenn Wood and Rob A. Rutenbar. FPGA routing and routability estimation via boolean satisfiability. In *FPGA*, pages 119–125, 1997.
- [15] Qiong Yu, Sandeep Badida, and Naveed Sherwani. Algorithmic aspects of three dimensional mcm routing. In *Proc. Design Automation Conf*, page 24.5, 1994.