Interconnect Synthesis for Lithography and Manufacturability in Deep Submicron Design

Sameer Pujari[1] Ryon M. Smey[1,2] Tan Yan[3]
Hannibal H. Madden[4] Patrick H. Madden[1,3]
State University of New York at Binghamton[1]
InternetCAD[2] University of Kitakyushu[3] AVS[4]

Abstract— Effective design for large VLSI systems requires abstraction; problems are simply too complex to be addressed directly. To obtain good results, it is necessary that the abstractions still capture the basic nature of the problem. Deep submicron lithography has placed new constraints on circuit layout; these constraints are frequently counterintuitive, and are hard to model with current design rules. When design tools ignore the constraints, there is a need for a great deal of "back end" work to fix violations–the difficulty of these fixes has resulted in a push towards very restrictive design rules.

To enable aggressive design without major violations, better abstractions are needed. In this paper, we focus on circuit interconnect, and develop a "straw man" approach to considering the lithography and manufacturability challenges of deep submicron design. We propose Mead-and-Conway style rules, and the concept of a "virtual layer" to separate mask-based constraints from silicon-based constraints. We also discuss a prototype routing tool that uses the virtual layer approach.

I. Introduction

In early technology generations, much of the "physical" nature of system design problems could be abstracted away, allowing a practical separation of logical function and physical implementation. In current technology, the dominance of interconnect delay on performance now means that even at very early stages of design, physical effects must be considered.

As technology scales further, lithography has become problematic. Advanced lithographic techniques are required for some of the smaller circuit features–masks used during manufacturing may not resemble the intended circuit structure. Not only must we consider the circuit structure that we desire; we also must consider the mask needed to realize it. Currently, the most significant modifications needed for manufacturing are limited to the polysilicon layers. For future generations, however, design automation tools that are oblivious to the manufacturing process will be unacceptable.

There is growing concern about this topic. Liebmann[13] presented a lithographers perspective to an audience primarily interested in physical design. This talk was followed by a call by Leung[12] for routing tools which could adjust to the changing technology, and a third talk by Kahng[9] that suggested that both design rules and routing tools needed to evolve. Clearly, there is considerable interest in tools that address the manufacturing process; calls for research in this area have been made for years (for example, [10, 14, 18, 20, 6]). There are also considerable benefits to directly considering manufacturing issues during design; Grobman[6] notes the following: *Even though it is not practical to come up with design rules to guarantee physical layout to be fully OPC complaint, it is possible to devise general rules so the physical design is more OPC friendly. This can drastically reduce the OPC development time, OPC correction run time, and the OPC output file size.*

In addition to the lithography issues are those of improving yield; some circuit structures have higher failure rates than others. By inserting a double-via versus a single via, an interconnect wire might have a slightly better probability of working correctly–if used across an entire chip, this can translate into fewer failures. Similarly, changes of wiring direction should be avoided if possible. As feature sizes scale, yield and performance become more sensitive to the subtle choices made by design tools.

Given the attention that lithography and manufacturability have received, and the importance of these topics to the entire semiconductor industry, one might expect a clear statement of what the major problems are, and what a "solution" should look like. In fact, very little information is available. In a panel discussion, Scheffer[19] suggested somewhat facetiously that the mafia or KGB should be hired to steal fabrication data from semicondutor manufacturers. While not as extreme, de Geus[3] has also made clear that design data is difficult to obtain. The semiconductor industry is intensely competative: those involved are reluctant to release any information that might help a rival.

These issues lead to a dilemma. Manufacturing processes must be considered for effective design–yet accurate information about the processes themeselves is not forthcoming. Further, it seems unlikely that this information will be available any time in the near future. As an attempt to resolve this dilemma, our research focus is on the development of an approach that allows design tools to consider manufacturing, without the need for manufacturers to make process details completely public.

In this paper, we first suggest a methodology for considering lithographic constraints during physical design. We also have a "straw man" formulation for considering manufacturing objectives during physical design. The "lambda rules" of Mead and Conway[17] greatly simplified the construction of physical design tools; to handle modern constraints, these rules must be extended. One of our objectives is to provide a level of abstraction that simplifies the constraints on earlier stages of design. Design automation tools perform optimization; if we wish to

consider new manufacturing issues effectively, we must clearly define what we wish to optimize and how we determine if one solution is better than another. In this paper, we also discuss a prototype approach to area routing, which can address the lithographic and manufacturing objectives.

To our knowledge, there are no generally accepted "metrics" to measure how manufacturable a design is; we hope to spark discussion on this area. The optimizations performed by our routing tool are based on discussions with colleagues in industry, and we gauge the performance of our approach by tracking the number of "undesirable" features in routings.

Our main interest is in the synthesis of circuit interconnect; at the transistor level, a great deal of effort can be put into the design of logic cells, allowing careful consideration of lithographic and manufacturability issues. Circuit interconnect, on the other hand, is essentially "custom," and there is a great deal of it: the amount of wiring needed for a large design can only be accomplished by an automated tool, and this tool must understand the new constraints if we are to have aggressive design.

II. Background and Previous Work

In earlier technology generations, the transition from the conceptions made by circuit designers into something that could be manufactured was relatively straight-forward. If a designer wished to have a square on a particular circuit layer, he or she would simply draw a square. A lithographic mask could be made from the drawing; the mask would contain a square–perhaps scaled, but of the same basic shape. Lithography and etching resulted in the square being realized as the designer intended.

As feature sizes have scaled down, the process has become more convoluted. Lithography introduces blurring; without modification, the corners of a square might become rounded. For very small squares, the rounding could be significant, with the desired feature being quite different from what was intended. Further, two closely spaced features might be realizable, but three might blur into eachother. The simple abstraction made by Mead and Conway fails to capture the complexity of the modern lithographic process.

Circuit design is no longer "what you see is what you get," and a designer must consider not only the properties of the circuit being designed, but also the nature of the mask needed to manufacture the circuit. To address this problem, the design rules and approach to interconnect design must change.

In this section, we discuss the manufacturing issues that are emerging. Very aggressive designs have encountered these challenges on the lowest layers (which require the smallest features). As sizes continue to scale, the challenges will be seen on a wider range of designs, and on more layers. This section concludes with a discussion of prior routing methods.

A. Manufacturing Issues

The manufacture of integrated circuits relies heavily on lithography and etching. We would suggest [25] as a good reference text. Desired circuit structures are first created (at a larger scale) on a transparent mask substrate; this process is time consuming, expensive, and requires lengthy verification and correction steps. Once a mask has been created, it can be used to expose a silicon wafer to laser light; a photosensitive coating on the silicon surface reacts to this light. After exposure, the desired structure can be created on the silicon surface by etching away material that has not been protected by the photosensitive coating.

With any lithographic process, however, there is unavoidable blurring of the image. In deep submicron design, this blurring can be substantial, and without lithographic "tricks," we would be unable to image features of the desired sizes and shapes.

Currently, circuit routing is performed without consideration of these issues. After the circuit design tools have completed their work, corrections and modifications are performed by lithography aware software to produce the image that will be placed onto a mask. This is a non-trivial process; in some cases, modifications to the circuit are necessary to make mask construction possible.

These corrections are most commonly applied to the gates in a design; these features are by far the smallest, and the most in need of "lithographic help." For future technology generations, the need for these corrections will be pervasive to most layers of interconnect, and design tools will need to consider them directly [13, 9].

Optical Proximity Correction (OPC) is a generic term for changes to the mask that cause the resulting structure on the chip to more closely resemble what a circuit designer intended. Common OPC features are illustrated in Figure 1. *Hammerheads and serifs* are added to the endpoints and corners of rectangles; without these, the corners can become "rounded," increasing the chance that the circuit will fail to operate as intended. Similarly, interior corners may have *cutouts.*

Some current fabrication processes require *line extensions*; the mask feature may need to extend beyond the location of a desired chip feature, to address the shortening of the line during lithography.

Subresolution Assist Features are small additions to a mask that increase the light level in a region. This additional light can give better line width control for desired structures, but the subresolution features themselves do not print.

These corrections have a significant impact on how circuits must be routed. Hammerheads and serifs from different wires may interact; simple spacing rules no longer apply. In some cases, the corrections may be optional (improving yield), while in other cases, the corrections may be absolutely necessary. Subresolution assist features introduce very complex constraints on route spacing; there may be minimum *and maximum* spacing between wires, with different spacings having different impacts on circuit yield.

"Fat wire"[12] constraints are also an example of how lithography is impacting circuit design. The minimum spacing for a wire may depend on the width of a nearby neighbor; if the neighbor is "fat," we may require substantially more space than if the neighbor is "thin."

While we do not address these issues in the current area routing prototype, we note that phase shifting masks[16, 13] complicate design to a great extent. We are not aware of any routing tool which considers this directly.

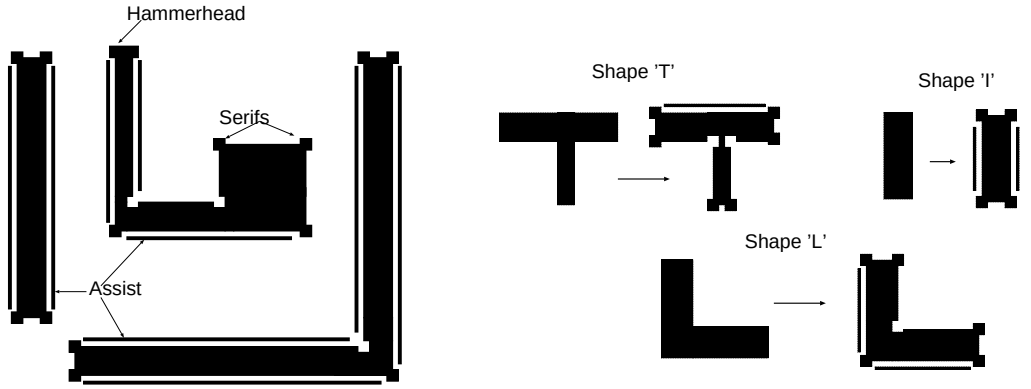

Fig. 1. Advanced lithographic processes require the introduction of a number of mask features; hammerheads, serifs, and subresolution assist features are becoming more common, and will be essential on future technology nodes.

B. Yield Enhancement

Beyond the lithographic constraints are new issues such as a strong preference for double vias; when possible, we may wish to insert rectangular vias to reduce resistance.

The double vias also serve to improve yield. In the manufacturing process, there are unavoidable variations–slight differences in the focus of the lithography, or the speed of etching, or the materials themselves, and introduce failures. A larger via is less susceptible to these variations.

There are similar issues with the lengths and widths of wires, bends in wires, non-preferred direction routing, and so forth. While a particular feature may be manufacturable, it might have better (or worse) failure rates than another type of feature.

Ideally, we would wish to have design automation tools which could optimize for yield–perhaps choosing to insert a via instead of bending a route, or increasing the width of a non-critical wire. Much of the data needed to perform this sort of optimization is proprietary; even if it were available, there is no accepted method for modeling it.

C. Circuit Routing

Circuit routing is a well studied area; normally, routing is decomposed into global and detail routing steps. Currently, global routing is performed by dividing the circuit area into a set of tiles, and then finding connections between tiles with repeated maze routing[15, 28, 29] or with a multi-commodity flow algorithm[21][2][1].

Global routing can be viewed as a somewhat abstract problem; a simple graph formulation is frequently sufficient. Area routing, on the other hand, is quite complex[24]. Even for relatively simple design rules, constructing a shortest path algorithm can be quite challenging[4]. Timing optimization also makes maze routing difficult[8]. As design rules become more complex, many have suggested that a move towards pattern routing or topological routing[23] may become more common.

Most area routing tools employ extensive maze routing (sometimes with a grid, to improve run times). Connections are made sequentially, with a "rip-up and reroute" process to eliminate blockages. This approach is compuationally expensive, and provides few guarantees about quality; it's popularity is due to it's success *in practice.*

The approach we consider here avoids maze routing for the reasons mentioned: simply put, considering emerging design constraints will become too difficult, and any router which uses maze routing extensively will likely be too slow to be practical. Instead, we employ pattern routing (simple combinationally-generated paths), with combinatorial optimization being used to select a set of routes that do not conflict.

Routing by a combinatorial methods is not new; for FPGA routing, use of satisfiability engines is well known[26], and similar techniques have been used on a small scale in integrated circuit routing. While a detailed description is not available, the "liquid router" used in X-architecture designs[27] apparently combines topological routing with combinatorial optimization.

III. Straw Man Methods and Metrics

Without question, something must be done. What that something is, however, is not yet clear. Some basic observations guide our proposal. First, rules must be developed to allow lithographic and manufacturing constraints to be considered: without these rules, design tools will operate in a blind fashion, causing significant problems. Second, these rules must accurately model the real constraints–if we are too conservative, we risk losing performance. Finally, the rules must be simple: if they are too complex, it will be difficult to construct design automation tools that support them, and there will be a significant run time penalty.

Our proposal for an appropriate method for considering lithography and manufacturing issues during interconnect synthesis is the following.

- Given a routing, features such as hammerheads, serifs, and sub-resolution assist features should be constructed using a simple rule-based approach. The rules can use a lambda-like formulation, and are applied based on the size and shape of the features. Note that these OPC features should not be included in the "final output" of the design automation tools; model-based methods used by lithographers are superior to rule-based methods, and the generated features should only be used to guide optimization. How this is to be accomplished is described below.

- Routing solutions should be evaluated on the traditonal metrics (wire length, delay), as well as the number and type of resolution enhancing features that can be added without creating violations.
- Third, we may also wish to track the number and type of "dangerous" constructions; single vias are less reliable than double vias, so we may wish to avoid them.

By using a lambda-like formulation, we can avoid creating tools that are directly tied to a particular foundary. The lithographic community may also be more willing to reveal "ballpark" values.

While it is unlikely that precise yield data will be made available in the near future, one might hope to know the failure rates for particular structures. If, for example, the failure rates for thin lines and for thin lines with subresolution assist features were known, a design automation tool could effectively evaluate the impact of adding the assist feature.

During routing, it would be extremely useful to know

- what is the yield impact if a particular feature is used?
- would an alternate routing increase or decrease yield?
- should a performance tradeoff be accepted to improve yield, or to reduce variability?

To our knowledge, metrics of this sort are not available, and we are not aware of any design automation tool that considers them directly. If this information were available, however, automation tools might be able to exploit this for significant improvements in yield and performance.

A. Virtual Layers

Lithographers have been critical of "OPC-modified" layouts; the features inserted by designers frequently cause a great deal of problem. To enable a designer to consider modern constraints without complicating the task for lithographers, we propose "virtual layers." A virtual layer is not actually fabricated; we use it to hold additional information about that design, related to desired "real" features placed on other layers. The proposed approach is illustrated in Figure 2. The desired silicon features are drawn by the circuit designer (or automated tools); rule-based OPC is applied to guide the physical design tools. Optionally, a designer may also mark portions of the design that are performance critical.

Additional virtual layers are created for each "real mask" layer. These layers hold features such as serifs and subresolution assist features; they enable a designer to plan for lithography, and to communicate design intent to the lithographer without making the problem more complex.

Lithographic constraints (and suggestions) are given as rules that apply to a pair of real and virtual layers. For example, if we wish to have a sharp corner on a small feature, we might need a serif; the serif shape can be placed on a virtual "metal 1 assist" layer, while the desired feature is on the "metal 1" layer. This combination allows the designer to more easily understand the spacing rules required because of lithography, and it provides the mask maker a clear indication that the designer truly needs a sharp corner. Obviously, the rules attached to virtual layers should not be viewed as hard constraints–they are based on simple rule-based lithography simulation, and not on physical models.

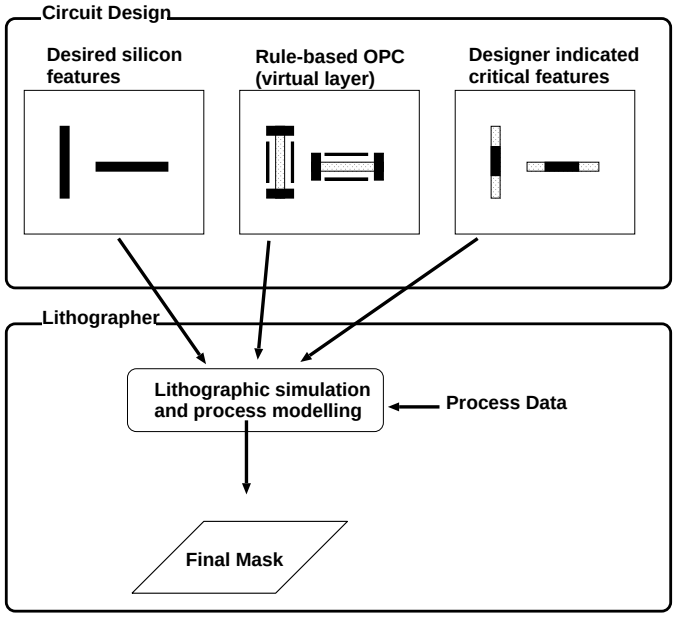

Fig. 2. Rather than having a single set of features defined for each layer, we propose their division into desired silicon shapes, rule-based OPC features, and indications of critical portions of the design. This information is specified by the circuit designer; the lithographer can then easily extract the designers intent, and can provide general lithographic guidelines without revealing process details.

We have developed primitive tools to allow manual design, and have simple rule-based methods to generate "recommended" modifications for the mask. We have also developed a prototype routing tool which uses the same methodology.

There are in fact many instances where it would be useful to provide additional "guidance" to the mask maker. Fill insertion, for example, can result in extremely large tape-out files, allowing representation of features that a designer would have very little concern over. By creating a "metal 1 fill layer," for example, a large region (which would in fact need many features) could be represented with a single polygon.

In many respects, the traditional flow after tape-out is quite literal; there is an attempt to fabricate *exactly* what has been drawn by the design tools. We suggest that by using a richer language to describe the design (with additional virtual layers), we can shrink the file sizes required, and also allow a designer to more accurately express intent.

The method of specifying areas critical performance has been pursued by industry groups. For example, a KLA-Tencor patent[5] describes "flagging" critical regions of a design. By doing so, the lithographer can know what portions of a design must be preserved, and what portions are free for modification (to simplify the lithography). Our "virtual layer" idea constrasts with this slightly, as we are attempting to directly model the constraints imposed by lithography, rather than marking specific features to preserve. The "virtual layer" approach allows a design tool to be more "lithography friendly." We feel these approaches

are complementary.

IV. A Prototype Tool

In this section, we describe a routing approach that employs virtual layers to directly consider the lithography constraints. It also supports weighting of routes so that we prefer constructions with better yield. The approach is based on our prior area routing tool[22]. The tool is currently being integrated with our mixed block placement[11] and global routing[7] work.

Rather than pursuing a traditional maze-routing based approach, we focus instead on combinatorial optimization. Our approach can be summarized by the following.

- For each connection, a number of routes are generated using correct-by-construction methods. Figure 3 illustrates some of the possible routings that can be produced with simple enumeration–at this stage we *ignore* any design rule violation with obstacles or other routes. Included with these routings are expected lithographic corrections, and estimates of the manufacturability of the features.
- We select a *subset* of the generated routes, to minimize the number of design rule violations, and to maximize the manufacturability. Our selection method employs a simple greedy heuristic; the underlying problem is in fact NP-Complete.
- After design rule violations have been eliminated, there may be many unconnected routes; a second set of routes is generated (again using correct-by-construction methods), but with a preference for locations that allow via alignment, or routing in unoccupied regions.
- The generation and elimination process proceeds for a user specified number of iterations. In practice, we find that some routes must be completed using a maze routing tool; a commercial tool which incorporates the basic approach described above obtains most routings through the combinatorial method, with completion by a sophisticated rip-up and reroute engine.

Motivation for the approach should be obvious: maze routing under the emerging design constraints is exceptionally difficult, and adding consideration into a maze router will be computationally expensive. By simply generating design-rule-correct routes, and then checking "can these two routes co-exist," we obtain a fast routing tool that can adapt to new technology constraints without needing to be overly conservative.

A. Terminology and an Example

To illustrate the process, we consider a small example with only a few nets. First, our constructive process generates a few possible routes for each net; we call each possible route a *candidate route*, and the set of candidate routes for a net a *bundle*. In our tool, the candidate routes included rule-based OPC features. This is illustrated in Figure 3. Next, our design rule checking tool identifies violations (and also preferred combinations) between pairs of candidate routes. This produces a constraint graph; the routing solution is obtained by removing some candidate routes, to optimize the number of completed connections, with a preference for solutions that simplify introduction of OPC features and utilize preferred wire spacings.

B. Route Generation

To support lithographic constraints, we made the following modifications to our area router. The first step in this conversion was the inclusion of OPC-enhanced routes into the pattern routing generation algorithms. The routes were generated using MOSIS design rules; hammerheads, serifs, and subresolution assist features were added using a rule-based approach.

By generating routes in this fashion, it is possible to store both the desired silicon features and the mask enhancements separately. Only the "intended" silicon features should be included in the router output; inserting "expected" OPC features only complicate the task of the lithographer.

C. Design Rule Checking

Our earlier area routing tool checked pairs of routes for violations (either too little spacing or intersections). These violations were used to construct a constraint graph used during optimization.

A second enhancement to the area routing tool was in the design rule checking step. Where previously we simply needed to check if features violated minimum spacing rules, we now consider both minimum and maximum spacings (to control routing pitch), and the types of violations generated.

Violations that occur due to OPC-related mask features coming too close are penalized less heavily than other types of violations. As OPC features may be optional, we assume that in some cases they can be eliminated (or a work-around can be constructed by the lithographic software).

When OPC features intersect, it may sometimes be *beneficial*. For example, a subresolution assist line may provide benefit to two desired features. The preferred and forbidden route spacings are directly related to the ability of lines to share subresolution assist lines.

Design rule checking encompassed the second stage of the required modifications. Here the DRC checker was modified to test to see if OPC features were creating design violations. As actual OPC spacing requirements are not publicly available, our routing tool uses constraints that seem reasonable from our survey of the existing literature. We consider serif size and spacing, assist feature sizes, spacing, and threshold lengths, hammerhead sizes, spacing, and threshold lengths, and non-square via shapes. These constraints are user-definable, and are applied on a per-layer basis.

With these parameters set, our design rule checker can find violations of several different types. Violations from optional features (such as subresolution assist segments) are less serious than basic spacing or intersection problems, and are treated differently by the routing solver.

D. Solver Modifications

Obtaining a routing with our approach requires finding a *subset* of possible routes; we have enhanced an existing solver in

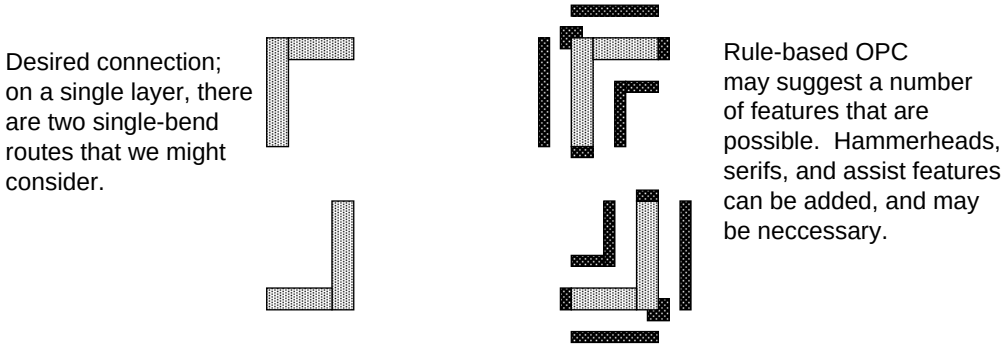

Fig. 3. Routing between a pair of points can be accomplished with different L-shaped bends (and on many different layers). Each variation requires different OPC features, via sizes and locations, etc. In our work, we use rule-based methods to determine what features are likely to be added to improve manufacturability.

the following way. Violations of desired silicon shapes are penalized heavily; when selecting a route, we seek to eliminate all "silicon" violations. Violations of OPC corrections, however, have only modest penalties; during the construction of a mask, the lithographer may choose to eliminate some corrections, as not all "desired" corrections are absolutely neccessary. Finally, we prefer routings that avoid manufacturability problems; excessive bends in a route, or excessive numbers of vias, for example, should be avoided.

Cost functions are in essence costs between two candidate routes from different bundles which gets stored on an edge of the constraint graph. A simplified cost function for the detail router is below:

$$cost(a,b) = overlap(a,b) * OVERLAPCOST + cost(a)$$

where the overlap function is function of the form

$$overlap(a,b) = \begin{cases} 1 \, if \, a,b \, have \, DRC \\ 0 \, otherwise \end{cases}$$

All OPC features are treated as optional features in this scheme, so the cost must be modified in such a way as to represent that optional nature. Employed for that purpose is a simple squashing function ensures the cost no removing the OPC features is always less than that of DRC penalties. It format is as follows:

$$squash(a,b) = \left(1 - \frac{1+opcretained(a,b)}{2+opctotal(a)}\right) OVERLAPCOST$$

Two additional pieces of information, opc_retained and opc_total, are easily obtained from the stage two modifications to the DRC process. Stated in straightforward terms, the opc_total variable is the total number of rectangles generated for the route by the OPC addition, and opc_retained is the opc_total minus the number of rectangles the modified DRC check for OPC features found fault with for route a to b existing simultaneously. The first term of the squashing function is an adjusted ratio between the number of OPC rectangle that are viable to the total number available and lies in the interval (0,1) non inclusive. Multiplying this term by the overlap cost leaves the squashing function always less than the overlap cost by definition. Based on the above squashing function the new cost function has the format:

$$cost(a,b) = overlap(a,b) * OVERLAPCOST + cost(a) + squash(a,b)$$

E. Implementation Details

Identification of violations is done on a layer-by-layer basis, and with computational geometry techniques. Thus, there is no need to check all pairs of features for conflicts, and the graph which represents the conflicts is relatively sparse.

We are currently moving from an implementation based on iterative deletion to one which employs dynamic programming. Details of this work will be presented at a later date.

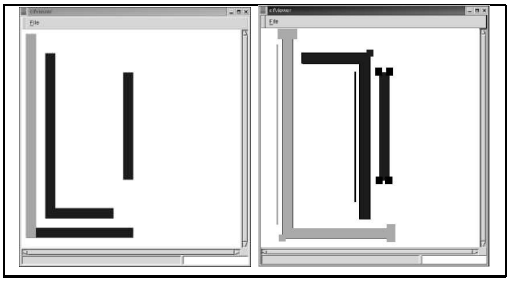

Fig. 4. Simple routings showing first a normal set of routes and the second shows the feature assists that are used to determine the routing. While the first routing is acceptable with traditional design rules, the introduction of OPC features into the optimization function creates a more "lithography friendly" design.

F. Preliminary Experiments

Support for lithographic constraints in our current routing tool is relatively limited; this is a work in progress, and we have constructed a few simple "proof of concept" tests. Figure 4 shows a simple example where a "traditional" routing solution on the left changes to a "lithography friendly" one on the right when OPC is considered. The basic routing approach can scale to larger problems–a variant of it is part of a current industrial routing tool. Note that the some OPC features (subresolution assist lines and serifs) for the second routing have been removed due to design rule violations; these areas can be flagged, possibly alerting the lithographer to "problem areas" that may be more difficult to handle.

Figure 5 shows a portion of a larger design. This problem is derived from a *Feng Shui 2.4*[11] placement of the MCNC

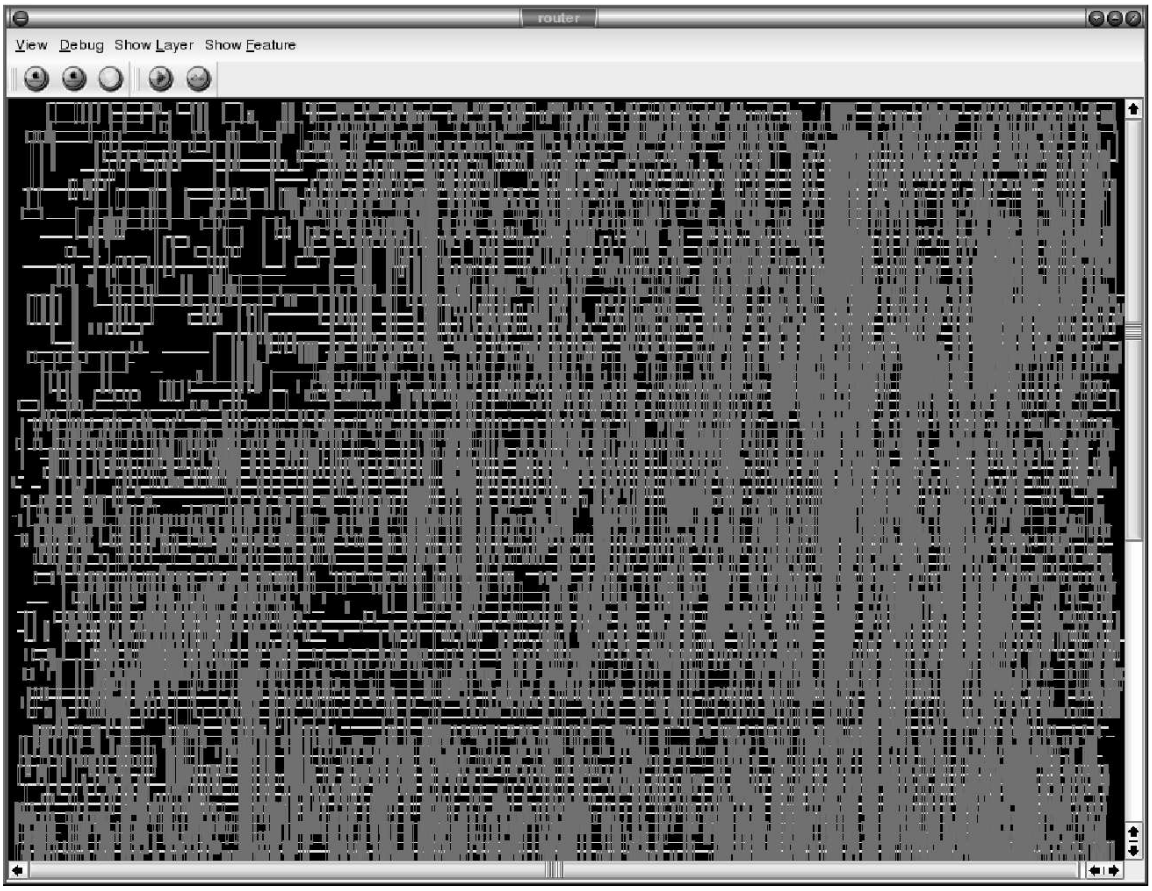

Fig. 5. Initial constraint graphs and routings for the MCNC benchmark "fract." All routes have OPC features (contained on a separate layer). A routing solution is found by selection of a subset of routes; removal of remaining violations is part of our current work.

benchmark "fract." As the design is small, we approach the problem flat, without a global routing step.

V. Summary and Conclusion

In this paper, we have proposed that simple rule based methods should be used to determine the types of lithographic features to be added during routing. In terms of methodology, we feel it is essential to support multiple "virtual layers" so that lithographic features inserted (to guide the designer and automation tools) can be easily distinguished from features actually desired on the silicon. Rule based methods are fast enough to be used during optimization, and can help make a design "lithography friendly." We also suggest that this is an area in need of a great deal more discussion.

We are quite interested in having more details about what causes chip failure, so that this information can be incorporated into the optimization step of our router. Normally, little is revealed by chip foundries; we would hope that it would not take massive production disasters to motivate the foundries to be more forthcoming with this information.

To experiment with "manufactuability driven routing," we have adapted an existing detail routing tool to optimize routes to support OPC features. Preliminary experiments are encouraging; the new tool clearly makes a design more compatible with advanced lithography compared to the "default" configuration.

As part of our current work, we are refining the routing tool to create more complex candidate routes, so that it is capable of handling very dense designs. We are also working with an industrial group on routing issues.

We plan to make the detail routing tool and our rule-based OPC software publicly available. There is an urgent need for greater lithographic support in design automation tools, and we hope that our contributions will spark more discussion and the development of metrics to be used in optimization.

Acknowledgements: This work was supported in part by NSF CCR 9988222, with additional support from an IBM Faculty Partnership Award, and NYSTAR-MDC. Some computing equipment was donated by Intel. We would like to thank Dr. Mark Lavin and Dr. Warren Grobman for helpful discussions.

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] R. C. Carden and C.-K. Cheng. A global router using an efficient approximate multicommodity flow. In *Proc. Design Automation Conf*, pages 316–321, 1991.

[3] P. Clarke and N. Mokhoff. Synopsys CEO calls for DFM cooperation; EE Times article http://eetimes.com/showarticle.jhtml?articleid=23904946, 2004.

[4] J. Cong, J. Fang, and K.-Y. Khoo. Via design rule consideration in multi-layer maze routing algorithms. In *Proc. Int. Symp. on Physical Design*, pages 214–218, 1999.

[5] L. A. Glasser, J. Ye, S.-T. Juang, D. S. Alles, and J. N. Wiley. US patent 6,529,621: Mechanismis for making and inspecting reticles, 2003.

[6] W. Grobman, M. Thompson, R. Wang, C. Yuan, R. Tian, and E. Demircan. Reticle enhancement technology: Implications and challenges for physical design. In *Proc. Design Automation Conf*, 2001.

[7] R. T. Hadsell and P. H. Madden. Improved global routing through congestion estimation. In *Proc. Design Automation Conf*, pages 28–31, 2003.

[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] A. B. Kahng. Research directions for coevolution of rules and routers. In *Proc. Int. Symp. on Physical Design*, pages 122–126, 2003.

[10] A. B. Kahng and Y. C. Pati. Subwavelength lithography and its potential impact on design and eda. In *Proc. Design Automation Conf*, 1999.

[11] A. Khatkhate, C. Li, A. R. Agnihotri, M. C. Yildiz, s. Ono, C.-K. Koh, and P. H. Madden. Recursive bisection based mixed block placement. In *Proc. Int. Symp. on Physical Design*, pages 84–89, 2004.

[12] H. K.-S. Leung. Advanced routing in changing technology landscape. In *Proc. Int. Symp. on Physical Design*, pages 118–121, 2003.

[13] L. W. Liebmann. Layout impact resolution enhancement techniques: Impediment or opportunity? In *Proc. Int. Symp. on Physical Design*, pages 110–117, 2003.

[14] L. W. Liebmann, S. M. Mansfield, A. K. Wong, M. A. Lavin, W. C. Leipold, and T. G. Dunham. TCAD development for lithography resolution enhancement. *IBM Journal of Research and Development*, 45(5):651–665, 2001.

[15] R. Linsker. An iterative-improvement penalty-function-driven wire routing system. *IBM Journal of Research and Development*, 28(5):613–624, September 1984.

[16] Y. Liu, A. Zakhor, and M. A. Zuniga. Computer-aided phase shift mask design with reduced complexity. *IEEE Trans. Semiconductor Manufacturing*, 9(2):170–181, May 1996.

[17] C. Mead and L. Conway. *Introduction to VLSI Systems*. Addison-Wesley, 1993.

[18] M. L. Rieger, J. P. Mayhew, and S. Panchapakesan. Layout design methodologies for sub-wavelength manufacturing. In *Proc. Design Automation Conf*, 2001.

[19] L. Scheffer. Nanometer design: Place your bets (panel). In *Proc. Design Automation Conf*, pages 546–547, 2003.

[20] F. M. Schellenberg, Luigi Capodieci, and Bob Socha. Adoption of opc and the impact on design and layout. In *Proc. Design Automation Conf*, 2001.

[21] E. Shragowitz and S. Keel. A global router based on a multicommodity flow model. *Integration, the VLSI Journal*, 5:3–16, 1987.

[22] R. M. Smey, W. P. Swartz, and P. H. Madden. Crosstalk reduction in area routing. In *Proc. Design, Automation and Test in Europe Conf.*, pages 862–867, 2003.

[23] J. Z. Su and W. W. Dai. Post-route optimization for improved yield using a rubber-band wiring model. In *Proc. Int. Conf. on Computer Aided Design*, pages 700–706, 1997.

[24] H.-P. Tseng, L. Scheffer, and C. Sechen. Timing and crosstalk driven area routing. In *Proc. Design Automation Conf*, pages 378–381, 1998.

[25] A. K.-K. Wong. *Resolution Enhancement Techniques in Optical Lithography*. SPIE - The International Society for Optical Engineering, 2001.

[26] R. G. Wood and R. A. Rutenbar. FPGA routing and routability estimation via boolean satisfiability. In *FPGA*, pages 119–125, 1997.

[27] X Initiative. Home page. http://www.xinitiative.org.

[28] Q. Yu, S. Badida, and N. Sherwani. Algorithmic aspects of three dimensional mcm routing. In *Proc. Design Automation Conf*, page 24.5, 1994.

[29] H. Zhou and D. F. Wong. Global routing with crosstalk constraints. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 18(11):1683–1688, November 1999.