

# CAD Computation for Manufacturability: Can We Save VLSI Technology from Itself?

Mark Lavin, Lars Liebmann,  
IBM Research Division  
IBM Corporation, Yorktown Heights NY 10598

**Abstract** -- Every 18 to 24 months, the areal density of VLSI doubles and the predicted date for the End Of CMOS Scaling is pushed out approximately 18 to 24 months. This rate of growth has been controlled mainly by the increasing capabilities of lithographic patterning. However, the rate of improvement of lithography systems in key physical parameters such as illumination wavelength has begun to slow. To make up the shortfall, lithography has increasingly turned to using CAD tools to transform the geometric shapes in designs into shapes on photomasks which have been compensated for systematic pattern-distorting effects. As the requirements for this compensation grow, however, it becomes increasingly difficult to hide in post-tapeout data preparation, and must be considered in back-end design tools and methodologies. We describe a number of the challenges to lithographic patterning, highlighting the factors that limit "physical scaling" and introduce the layout-to-mask shape transformations that compensate for these limitations. We describe the implementation of these transformations in general-purpose and specialized CAD tools, pointing out challenges like growth of computation effort. Finally, we describe how limitations of post-tapeout compensation drive the need for "litho-aware" physical design tools, showing examples in cell design, place-and-route, and layout migration.

## I. INTRODUCTION

Integrated Microelectronics provides the means to design digital and analog electronic circuits that implement an enormous variety of complex function and to manufacture those circuits with great productivity and remarkably low cost. Users demand that the variety and complexity of the circuits grow while the productivity increases and cost declines. As this demand stresses the physical principles underlying these circuits and their fabrication, strain starts to appear. This paper looks at one such strain: the increasing distortion of fabricated material geometries w.r.t. their corresponding design

geometries; section II reviews IC fabrication processes and lists a number of the sources of distortion. Increasingly, improvements in materials and fabrication processes are supplemented by computations that compensate for the limitations of the material and fabrication processes by systematically transforming the designed shapes into the ones on lithographic photomasks. In some cases, this compensation can be performed in "post tape-out" computations invisible to the design process, employing CAD tools used originally for verifying physical designs; section III describes these computations. As the capabilities of physical materials and processes are pushed further, however, it's apparent that the compensations can no longer be hidden from the design process; section IV presents examples of this emerging class of "in-design" computations. Both the post tape-out and in-design computations present significant challenges to developers of CAD algorithms and systems; these are discussed in section V.

## II. THE PROBLEM: IC FABRICATION PROCESSES DISTORT SHAPES

The flexibility, complexity, and productivity of Integrated Microelectronics derives largely from *photolithographic patterning*, a set of optical and chemical processes for imposing specified geometric structures on materials. The precision and accuracy of those structures is limited by the patterning processes, and this is significant because those geometric distortions limit (or completely negate) the function and performance of the resulting circuits. In this section, we'll briefly review the flow of processes used to fabricate integrated circuits, then highlight the sources of shape distortion along that flow.

### A. IC Fabrication Process Flow

The fabrication of one material layer of an integrated circuit (IC), e.g. the polysilicon gate layer, consists of the following steps:

- At the completion of the design process, the final chip layout, a set of geometric shapes on specified material or mask levels, is *taped-out*, converted to a data stream format, typically GDSII, and transferred to the IC manufacturer.
- These layout data are converted by *mask-fracturing* software into tool-specific formats (e.g., MEBES) then used by *mask-writing tools* to create geometric patterns on *photomasks*, objects similar to photographic negatives.
- *Wafer-steppers*, specialized optical instruments, project light through the photomasks onto the surface of silicon wafers that are coated with light-sensitive *photoresist* material.
- Chemical *development processes* remove the resist material in the pattern transferred by light from the photomask.
- The patterned resist material acts as a template for either adding material to the semiconductor wafer through *deposition* or *implantation*, or removing it through *etching*.

For the example of patterning polysilicon gate material, fabrication begins by depositing a uniform layer of polycrystalline silicon on the semiconductor wafer (which has already been processed to produce active areas and isolation regions), and a layer of photoresist on top of that. “Poly” shapes in the design are transferred to opaque areas on the photomask. When the wafer is exposed through the mask, the PC shapes correspond to non-exposed areas of photoresist; these areas remain when the resist is developed. Next, the wafer is subject to reactive ion etching (RIE), but the regions covered by photoresist, corresponding to PC shapes, are not removed, and remain as small “wires” and “pads” after the photoresist is removed. Ideally, the shapes and dimensions of the fabricated polysilicon shapes exactly matches the Poly shapes in the taped-out layout, but physical reality intervenes.

### B. Shape Distortions due to Mask Making

Mask making has a unique role in lithographic patterning, being the one step where data are converted into physical structure; it is also a fabrication process unto itself, containing many of the same steps as chip fabrication. Mask making directly contributes several forms of shape distortions:

- Across-mask dimensional variations due to variations in materials and non-uniformities in mask fabrication processes.
- Local, pattern-dependent dimensional variations due to proximity effects such as electron beam backscatter.

- Corner rounding

Although these dimensional variations are limited in magnitude (few 10’s of nanometers) and mitigated by the fact that most photomasks are constructed at four times the intended wafer dimensions, they can have strong effects on the small variations in mask shapes that are introduced in the process of compensation for lithographic and other wafer fabrication effects.

Mask making also affects wafer shape distortion indirectly: limitations in mask fabrication processes and automated optical inspection introduce the need for constraints on minimum feature sizes on masks. These constraints limit the effectiveness of applications that compensate for wafer shape distortion, discussed below.

### C. Shape Distortions due to Lithographic effects

The designed layout can be represented by a two-valued characteristic function  $D(x,y)$  that is 1 where covered by a designed shape and 0 elsewhere. The photomask can be viewed as a two-valued function  $M(x,y)$  that is opaque or clear at each point, and ideally a replica of  $D(x,y)$ , ignoring the distorting effects of mask-making just discussed. However, due to effects of diffraction, the intensity of light projected by the wafer stepper through a given mask onto the surface of the photoresist is a continuous function  $I(x,y)$  that is both “smoothed” compared with  $M(x,y)$  and also distorted because  $I(x,y)$  is a complex combination of  $M(x_i,y_i)$  at many points in the vicinity of  $(x,y)$ .  $I(x,y)$  can be thought of as a low-pass filtered version of  $M(x,y)$ , where the spatial frequency cutoff is defined by the Rayleigh equation  $R = k_1 \frac{\lambda}{NA}$ , where  $R$  is the minimum resolvable line/space pair,  $\lambda$  and  $NA$  are the wavelength and numerical aperture (similar to f-stop) of the optical stepper, and  $k_1$  is a constant with a theoretical minimum of 0.25. As this factor decreases with successive technology generations, the shape distortion increases, as shown by the cross shape (mask) vs. the succession of four-fold blobs (wafer) in Figure 1.

These *optical proximity effects* are compounded by effects of the chemical and physical processes in which photoresist materials react to light and are eventually transformed into regions where resist is either removed or remains. These effects, which may distort resist shape dimensions by 10’s of nanometers (a substantial fraction of the nominal dimensions for minimum features in leading edge technologies) are further extended because of variations in materials (e.g., resist thickness) and process conditions (variations in intensity and focus across the field projected by the wafer stepper). In

addition, there are effects due to wafer-scale and wafer-to-wafer variations.

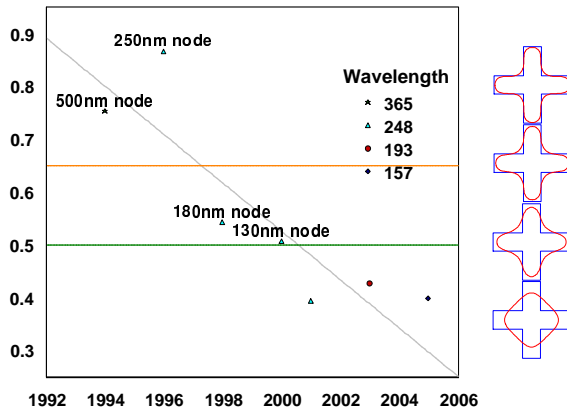


Figure 1: Rayleigh Constant for various Technologies

#### D. Shape Distortions due to Post-lithographic effects

For many of the material layers of integrated circuits, the next patterning step is transferring the shapes in the resist into the underlying material by *etching*, which removes material not covered by the resist. This transfer introduces several additional shape distortions:

- Uniform increases or decreases of shape dimension by non-directional reactions (sidewall etching or undercutting).
- Local pattern-dependent effects (“microloading”).
- Longer range effects due to differing overall pattern densities (fraction of underlying material that is removed).

Another source of shape distortions, albeit in the direction perpendicular to the wafer plane, comes from the processes used to *planarize* material layers, currently a combination of *reactive ion etching* (RIE) and *chemical-mechanical polishing* (CMP). These effects can alter the thickness of dielectric or conducting materials by 10’s of nanometers and, like all the effects listed above, are pattern-dependent. In addition to affecting the electrical properties of the planarized materials -- capacitance and resistance -- they indirectly distort shapes on subsequently fabricated layers by altering the relation of the material surface to the wafer stepper’s focal plane.

#### E. Problems caused by Shape Distortions

The shape distortions described above can affect circuit function, performance and reliability in a number of ways:

- Extreme variations can directly cause loss of connectivity or accidental connection (opens and shorts) or less directly increase susceptibility to spot defects.
- Changes of shape and inter-shape space dimensions can affect circuit parameters. The most important of these is the dependence of FET channel length  $L_{eff}$  on width of polysilicon lines over active area. Dimensional variations affect resistance of conductors and capacitance between adjacent conductors. These variations, in turn, can affect the behavior of circuit elements, limiting operating frequency or causing outright logic failures.

These effects are compounded by the dependence of shape distortions on process variations. Thus, although circuits manufactured at nominal conditions may function correctly at satisfactory performance, circuits manufactured under some variations of process may not perform adequately, if at all, effectively limiting the yield of the manufacturing process.

#### F. A solution approach: Process-based Design Rules

One solution to problems caused by process-induced shape distortions is to define *design rules*, constraints on the geometric properties and relations of shapes that can appear in a design for a given fabrication process; designs that satisfy these constraints are considered *manufacturable*, i.e., the fabricated circuits are predicted to function at satisfactory performance with sufficient yield despite the shape distortions that could result under the expected range of process variations.

Design rules are defined by analysis that: Computes the sensitivity of circuit function and performance to shape distortions; estimates the amount and type of shape distortions due to process effects as described above, in addition to other effects like layer-to-layer misalignment; determines the amount of “guard-band” for various shape dimensions and intershape relationships under which the probability of correct function is obtained using the sensitivities and variabilities.

Design rules, which include constraints like minimum width and spacing of features on specified layers, minimum overlap between features on different layers, are defined as part of the formal specification of a semiconductor technology. Design rules are checked by CAD programs for layout verification and shape processing: IBM’s Niagara, Mentor Graphics Corp. Calibre®, Cadence Design Systems Assura™, or Synopsis, Inc. Hercules-II, and the results are typically presented in a graphical user interface to designers to guide correction of errors.

### 3. POST-PROCESSING SOLUTIONS

Design rules reduce the complex relationship between processes, devices, electrical properties and circuit function and performance to a relatively small set of simple constraints on design geometries. However, there are several drawbacks to this approach: First, the more accurately the design rules reflect the true capabilities of the fabrication processes (i.e., they are not “safe-sided”) the more the resulting designs reflect characteristics of those specific processes, so that as the processes change, layouts must be redesigned. To address this problem it has been common practice to compensate for these changes by introducing simple post-design adjustments to designed shapes, e.g., expanding all shapes on certain levels uniformly in the case where a new etching technique having different lateral “bias” is introduced.

However, the effectiveness of these simple biases is limited: although many of the design rules are simple to state, check and correct independently, as process complexity increases, the rules grow more complex, as well. The alternative, which has been practiced more frequently as IC minimum feature sizes dropped below 1 micron, has been to transfer some of this complexity into more general *post-processing applications* that transform the “as-designed” shapes  $D(x,y)$  into shapes on the photomask  $M'(x,y)$  that, when fabricated, yield wafer shapes less distorted w.r.t.  $D(x,y)$

#### A. Rule-based Proximity Correction (OPC)

One of the most familiar process-induced shape distortions is “nested/isolated bias”, the phenomenon that the linewidth variation of shapes depends both their nominal width and their spacing (“nested” refers to shapes at minimum spacing). This dependence can be determined empirically -- by measuring the on-wafer width of lines with known on-mask width and spacing -- or by simulation using physically-based mathematical models.

Given the simple relationship between wafer linewidth and mask linewidth and spacing it is usually feasible to determine a functional rule that describes the inverse relationship, that is, given a desired width and spacing, what line width should appear on the mask or equivalently what “correction bias” should be applied to designed shapes, as shown in Figure 2.

This bias can be applied by shape processing programs of the kind used for checking design rules (which involve measuring local width and spacing). These techniques can be extended to variations that involve

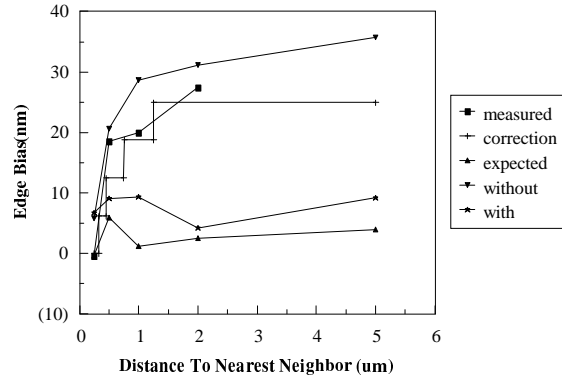


Figure 2: Rule-based Optical Proximity Correction

more pattern parameters than width and spacing, e.g., to addition of line end anchors to compensate for line end shortening and corner serifs for reducing corner-rounding effects. However, the effectiveness of these rule-based techniques diminishes when they’re applied to more complex 2D interactions, motivating the use of iterative *model-based correction* described below.

#### B. Density Correction (Filling, Slotting)

A number of the post-optical shape distortion effects, including those in etching and planarization, depend on *shape density* (fraction of area covered by shapes) at several different size scales. For this reason, technologies that use such processes often include design rules that constrain minimum and maximum shape density. In general, it is more difficult to modify design structure to satisfy density constraints than it is to increase or decrease local density through addition of non-functional shapes (“fill”) or removal of non-required parts of functional shapes (“slotting”) used separately or in combination

This compensation can be implemented simply (but possibly not efficiently) by using conventional layout checking or shape processing programs to create a virtual array of *fill shapes*, discarding those too close to functional shapes. To reduce resulting data volume, a number of techniques have been developed for compressing the representation using hierarchical design structures[1].

One operational difficulty distinguishes filling and slotting from other shape post-processing shape compensations: They result in structures on the wafer different from the original design. Since these structures may affect parasitic electrical properties, it is necessary to carefully characterize the effect of fill and in some cases, to compute parasitics taking fill into account and using this information to confirm timing closure.

### C. Model-based Proximity Correction

Model-based Proximity Correction addresses the same problem as rule-based OPC described above, using a different approach that overcomes some of its limitations for 2D patterns. Instead of directly computing the corrections to be applied, based on simple, local, one-dimensional features like width and spacing, MBOPC iteratively modifies the mask shape by checking the agreement between the as-designed shapes and process model simulations of the wafer shapes that the current mask shape would produce.

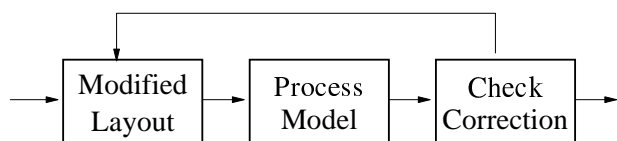


Figure 3: Model-based Optical Proximity Correction

The simulation is based on mathematical models that compute local, two-dimensional integral properties using an approximation to first-principle physical models [2] [3], augmented with “non-physical” degrees of freedom that can be used to fit the models to empirically measured data. In order to address side-constraints such as the *mask rules* described above, model-based OPC applications must extend the basic simulate-correct-iterate mechanism. It may also be desirable to introduce additional constraints to reduce the amount of computation and to limit output mask shape data volume.

### D. Sub-resolution Assist Features

Rule-based and model-based OPC can generate mask shapes that will result in wafer shapes that more closely match the as-designed shapes, *provided the process conditions match those from which the correction rules or models were derived*. However, OPC does not reduce the amount of shape distortion that occurs because of process variations, e.g., changes in dose and focus. However, several techniques can reduce the sensitivity of shape distortions to process variations.

One of these is the addition of *sub-resolution assist features*. These are narrow “outriggers” that do not themselves image on the wafer shown in Figure 4, but which affect the images of the intended features, reducing the amount by which edge locations vary with

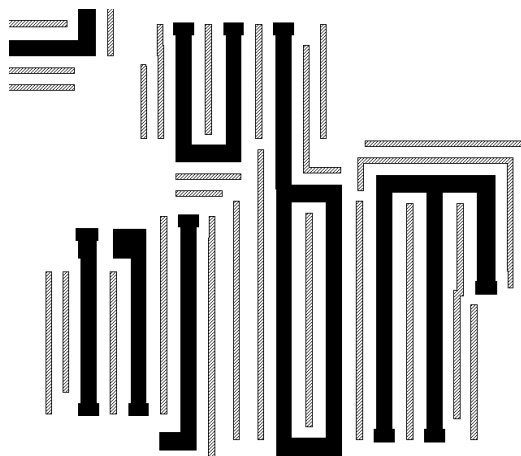


Figure 4: Subresolution Assist Features

changes in focus and dose, increasing the *process window*, as shown in Figure 5.

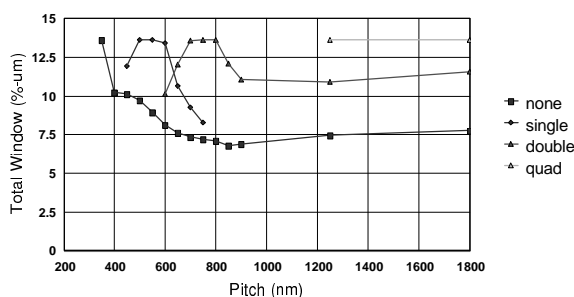


Figure 5: Increase of Process Window with Assists

Algorithms to generate assist features are straightforward on the surface: Rule tables like those in rule-based OPC are used to initially place “raw” assist features. These are subjected to a series of “cleaning” steps to remove violations of mask rules, and to ensure that there are no configurations of assist features that might accidentally “print” under extreme process variations. In detail, these cleaning steps are quite complicated, since the different constraints may conflict, and care must be taken not to clean the raw assists so much that they no longer provide assistance.

### E. Alternating Phase Shift Masks

In 1982, Marc Levenson proposed the technique of alternating phase shift mask (altPSM) lithography[4], in which binary chrome-on-glass (COG) photomasks are modified so that transparent regions are etched to impose several possible phase shifts on light passing through them. The resulting constructive and destructive

interference effectively doubles the spatial frequency of

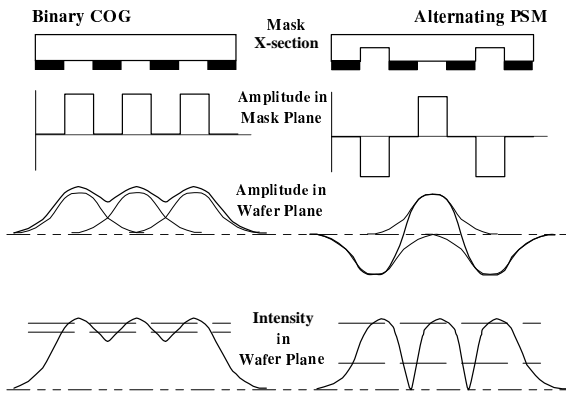


Figure 6: Alternating Phase Shift Mask Principle

patterns that can be printed, and reduces the variability of narrow features with respect to process variations.

Post-processing to generate altPSM is considerably more complex than techniques described above: the patterns of the phase shifting regions do not have the same topology as the designed features (compounded by the need for a second *trim* or *block* mask in most altPSM techniques), although the geometric operations to generate them are within the scope of layout verification or shape-processing programs. AltPSM mask generation requires an additional operation, the ability to “color” phase shift regions (assign one of several phase shift values) so that there are phase differences across all critical dimension designed features[5].

As with SRAF generation, while the basic flow of altPSM generation is straightforward -- identify critical designed features, generate phase shifters, clean mask rule errors, color phases -- the devil is in the details, particularly in the “clean mask rule errors” step. In addition, altSPM generation introduces a new problem: phase coloring conflicts. Configurations like T’s and “belt buckles” result in phase regions that cannot be consistently colored. A number of approaches have been proposed to identify such conflicts and automatically eliminate them[6].

#### F. Challenges of Post-processing

Post-processing uses shape-processing CAD tools to maintain the relatively simple design rule interface between process and design. As technologies and fabrication processes grow more complex, it stresses the capabilities of those tools and the postprocessing applications: First, in order to compensate for more complex shape distortions, additional functionality is required; an obvious example is the introduction of “simulation

engines” based on convolution into tools used for model-based OPC. Second, the growth of design complexity (increasing component counts, increasing numbers of mask levels) causes at least proportional growth in the amount of computation. Although this is balanced by increases in computer speed and capacity, there is a further compounding factor, the growth of shape interaction radius: As we model and compensate for more complex physical effects, the range over which mask shapes influence each other increases from one minimum line/space to several (up to 4 line/space pairs for 90nm technology). This increases computation directly by a factor of 16, and there is a secondary effect that may increase computation and result data size even more: the “flattening” effect that expands data that were originally “compressed” by hierarchical layout representation.

Finally, as the functional and performance requirements for post-processing tools and applications grow along with the pressures to reduce software development time, there is a clear need for tools and methodologies to test and validate the post-processing software **before** the results are committed to expensive hardware.

#### IV. SOLUTIONS THAT ARE PART OF THE DESIGN PROCESS

The problem with the post-processing solution to process-induced shape distortion is that instead of requiring designs to be “manufacturable”, they must be “post-processable” and “manufacturable after post-processing”. This leads to the dilemma where the post-processor application developer tells the designer “we can’t post process your design” and the designer asks “what do I change to fix it?”.

The obvious solution is to extend the design rules by somehow composing the manufacturability constraints with the requirements of the post-processing application (e.g. phase colorability). However, there is no formal procedure for doing this, so the usual approach is to add design rules or modify the bounds on existing rules until post-processability and manufacturability are ensured. In practice, however, the resulting design rules may be so conservative that the intended benefits (e.g., area density) of the new technology is lost.

The obvious solution to that problem is to make the design process, tools and methodologies “manufacturing-aware”; the difficulty with that is to determine what it means. Here are several possibilities:

##### A. Use the Postprocessor as the Checker

One way to avoid post-processability design rules is to make the post-processing program part of the design environment, launching it from an interactive layout design session and reporting its results there like conventional design rule checking programs. The key to this is that the post-processor must have some ability to localize the regions where conflicts occur: for altPSM, the obvious case is where phase conflicts occur. For SRAF generation, this could mean highlighting parts of critical features where assist shapes could not be placed because of conflicts with other assists or designed features.

A study[7] investigated the feasibility of this approach for using altPSM for 175nm node logic by introducing an altPSM post-processing program into the design environment used for several production chips. The finding was that designers accepted the use of the post-processor as checker, and quickly developed the ability to fix problems that the post-processor highlighted. Productivity was also improved by adapting the MASH tool [8] to automatically modify designs into “phase compliance”, using error shapes produced by the altPSM post-processor program to add constraints processed by MASH’s minimum perturbation compactor.

### *B. Use the Simulator as a Quantitative Checker*

Although installing postprocessing applications in the design environment eliminates the need for “post-processability rules”, it still has the binary “pass/fail” aspect of traditional design rules, providing little information or incentive for improving overall manufacturability. This might be addressed by adapting the *critical area* approach used in defect-limited yield analysis [9].

Instead of abstracting the process effects and variations into a set of rules and threshold, we could run the process “forward”, taking the as-designed, as-best-corrected shapes, simulating the process-distorted wafer shapes and computing simulated values for physical features that determine circuit behavior, such as  $L_{\text{eff}}$ , contact area, interconnect resistance, etc. The resulting values could be compared with electrical constraints or further simulated to verify correct time. By repeating this simulation and analysis over the range of expected process variations, including dose, focus and overlay, the circuit-limited yield (fraction of manufactured circuits satisfying performance and other constraints) could be estimate. In addition, it may be possible to localize hot spots where components go out of spec. Outside a narrow process window, and reflect these back to the designer for correction.

In practice, this approach would require significant computation and would thus be limited to small, frequently-used structures like SRAM cells and FPGA elements.

### *C. Add Postprocessability Constraints to the Layout Elements and Composition*

A third approach is to divide the problem hierarchically: Use a combination of traditional design rules, postprocessing programs, and simulation-based optimization to create manufacturable small elements, e.g., standard cells. In order to create manufacturable full chips, we must add *composability constraints* on the elements. For example in the altPSM study, three such constraints were suggested to ensure that arbitrarily-placed combinations of standard cells were altPSM-compliant (assuming only 1-dimension interactions): Extend the width of elements so that no color conflict could occur between adjacent element’s outermost phase shifter shapes; require all elements to start and end with the same shifter phase, or provide for both “standard” and “color-reversed” copies of each element in the library, and require the placement program to choose the appropriate parity.

Such constraints appear to address some of the problems of composability at acceptable cost (area and design effort) because they exploit existing constraints (row-based placement, element widths in discrete steps). It is speculated that global routing programs, which operate on a discrete grid of channels and nodes, would be amenable to such manufacturability-driven constraints.

In addition to being acceptable to both designers and design processes, these constraints are pleasing to lithographers and process developers because they reduce the diversity of geometries that processes have to reproduce.

## V. CONCLUSIONS

Requirements for Integrated Microelectronics are outstripping the capability of physical systems and materials that implement them. Increasingly, this shortfall can be reduced through the use of computer-aided design tools and design methodologies.

### *A. Implications for Design through Manufacturing Flow*

It is inevitable that the wall between Design and Manufacturing will have to come down post-tapeout

processing alone won't address process requirements, and satisfying traditional design rules will not assure manufacturability.

As described in Section IV, there are a number of approaches to breaching the wall. This is complicated by the fact at any given time there is not just "the process" but a range of processes, from mature to barely experimental, yet all must yield manufacturing result for the cycle to proceed. Furthermore, for a given process, there are multiple, divergent design styles -- DRAM, SRAM, ASIC, semi- and full-custom logic -- and all place varying priorities on development time vs. performance vs. chip density etc. This may mean that there are multiple flows and possibly multiple tool sets used during design and post-tapeout to ensure manufacturability.

The problem is further complicated by the fact that different fabricators will adopt different processes, possibly implying different design rules that require different tools for checking and compliance. For the moment, no single technology route promises convergence in the future.

#### *B. Implications for CAD Tools*

Increasing demands on processes will require increasing complexity of post-design applications to compensate for process-induced shape distortions. In addition, the computation cost grows more than linearly with the complexity of the designs that are being processed. These factors drive increases in computation requirements that outstrip improvement in uni-processor speed. This implies the need for algorithms that scale more gracefully with problem size and that can take advantage of multicomputer systems, both tightly- and loosely-coupled. Needs for more function and performance in turn drive the need for more rigorous validation of based software and applications used in post-processing mode, since these are often the last stage before designs (and errors) are committed to physical hardware.

For in-design tools, in addition to the problems of diversity in design flows, checking of/optimization for manufacturability must coexist with a number of other "directives" constraining design, including timing, power, area, noise, and power, increasing the overall amount of computation and the number of potential tool to tool interactions.

#### *C. Implications for Technology Development*

The increasing reliance of processes on CAD computation and the closer coupling between design and

manufacture will also impose constraints on the definition and development of new technology. Technology capability will have to be evaluated not only in terms of roadmap numbers like minimum feature size or pitch, but rather by the manufacturable and *designable* capability, i.e., what chip density can be achieved with acceptable design cost and time?

As Design starts to take on the cost of Manufacturing's process complexity, there will be strong pressure to support technologies on both sides of the wall that can lead to reuse of designs while processes evolve.

#### REFERENCES

- [1] D. Chesebro, Y. Kim, M. Lavin, D. Maynard, "Efficient Generation of Fill Shapes for Chips and Packages", U. S. Patent 5,636,133, June, 1997.
- [2] Y. C. Pati, A. A. Ghazanfarian, R. F. Pease, "Exploiting Structure in Fast Aerial Image Computation for IC Patterns", IEEE Trans. Semi. Mfg., Vol. 10, No. 1, pp. 62-74, Feb, 1997.
- [3] T. Brunner and R. Ferguson, "Approximate Models for Resist Process Effects", Proc. SPIE Vol 2726, 1996.
- [4] M. Levenson, N. Viswanathan and R. Simpson, "Improving resolution in photolithography with a phase-shifting mask", IEEE Trans. Electr. Devices, Vol. 29, pp1828-1836, Dec. 1982.
- [5] Y. Kim, M. Lavin, L. Liebmann, G. Weinert, "Use of Net Coloring for Automatic Generation of Phase Shift Masks", U. S. Patent 5,883,813, Mar. 1999.
- [6] T. Wass, et al., "Automatic Generation of Phase Shift Mask Layouts," Microelectronic Engineering, Vol. 23, pp. 139-142, 1994.
- [7] L. Liebmann, J. Lund, F.-L. Heng, I. Graur, "Enabling Alternating Phase Shifted Mask Designs for a Full Logic Gate Level: Design Rules and Design Rule Checking", Proc. DAC, 2001.
- [8] F.-L. Heng, J. Lund, L. Liebmann, "Application of Automated Design Migration to Alternating PSM Design", ISPD, 2001.
- [9] C. H. Stapper, On yield, fault distributions and clustering of particles, IBM J. Res. Devel., 30: 326-338, 1986.