Automated and integrated mask generation from a CAD constructed 3D model

Richard L. Schiek and Rodney C. Schmidt

rlschie@sandia.gov rcschmi@sandia.gov
Computational Sciences Department
Sandia National Laboratories, Albuquerque, NM, USA.

ABSTRACT

We have developed and implemented a method which given a three-dimensional object can infer from topology the two-dimensional masks needed to produce that object with surface micromachining. This design tool calculates the two-dimensional mask set required to produce a given three-dimensional model by investigating the vertical topology to the model. The 3D model is first separated into bodies that are non-intersecting, made from different materials or only linked through a ground plane. Next, for each body unique horizontal cross sections are located and arranged into a tree based on their topological relationship. A branch-wise search of the tree uncovers locations where deposition boundaries must lie and identifies candidate masks creating a generic mask set for the 3D model. Finally, in the last step specific process requirements are considered that may constrain the generic mask set.

Keywords: topology, mask, mems, design

1 DESIGN ISSUES

Designing a device for production by silicon micromachining is very different from macro-scale mechanical design. In the macro-scale it is often sufficient for a designer to create a 3D model of their device, which a design program then translates into the tool paths needed for production. For a silicon micromachined device however, the designer must create a set of process specific, lithographic masks needed to fabricate the device. Because masks are dependent on the process in which they are used and can have complex dependency interactions within a production system, creation of the masks is a significant challenge to innovative device design and the manufacture of a device on multiple processes.

Earlier efforts on this problem have leveraged existing technology in process simulators, *i.e.* programs which when supplied with a mask set for a given process can simulate fabrication from those masks. Typically, this approach uses a trial mask set to produce a 3D object that is then compared to the desired object. Differences between the two objects are used to alter the trial mask set and then the process is repeated until a mask set is found which correctly produces the desired part. When coupled with a sophisticated optimization scheme, this approach works well for anisotropic etching processes. [1] Being computationally intensive however, optimization trial masks through a process simulator has yet to produce masks for complex, multi-layer surface micromachined devices. Another approach starts from a 3D model that is annotated with data which describes when in the process each section of it will be made and from each annotated section a mask is derived. [2] More recently progress has been made on a geometric approach where a 3D model is interrogated for features that can be made via surface micromachining, and a mask set is derived for these features. [3] While promising, these techniques cannot produce masks for specific processes nor handle isotropic etching processes such as wet etches.

2 CROSS SECTIONS

Surface micromachining builds a MEMS device with the successive deposition and controlled etching of materials on a silicon substrate. Thus, the horizontal cross sections of a device can be used to identify the masks. Given a body, let z represent the scaler distance from a reference ground plane and let $C(z)$ denote the cross section of a body at the height z. The function $C(z)$ is not necessarily a continuous function of height as a part with exactly vertical sides will create discontinuities in $C(z)$ when the cross section changes.

While a cross section itself is infinitely thin, one can identify a range of heights within which a given cross section is constant. Thus, if one defines $\mathcal{C}$ as a constant cross section, one can then write:

$$P_i = C(z_i) : C(z) = \mathcal{C} \ \forall \ z \in [z_i, z_{i+1}) \qquad (1)$$

In defining P_i, one has implicitly subdivided the z domain into intervals within which a given cross section is constant. Since $C(z)$ may not be continuous, the range of heights within which $C(z)$ is constant cannot easily be defined as closed. Thus a range of acceptable z values can be written as either $[z_i, z_{i+1})$ or $(z_i, z_{i+1}]$ and the sequence z_i may be increasing or decreasing. In this analysis, it is assumed that the range of allowed heights is traversed from top to bottom implying $z_i > z_{i+1}$ and that any discontinuities in $C(z)$ are placed at the lower height yielding the closure defined in equation 1. Now, given P_i a set can defined as follows:

$$U = \{P_1, P_2, ...P_N\} \tag{2}$$

This defines U as the sequential set of all unique cross sections for a given body. Note that unique here only implies that P_i is not equivalent to either P_{i-1} or P_{i+1}, *i.e.* unique relative to ones neighbors. With a notation on hand to describe a body's cross sections and where they arise, attention next will be directed to organizing the cross sections into a useful topology tree.

2.1 Topology Graph Analysis

In general, a given cross section may contain multiple subcomponents, islands or lumps. To account for this one can expand the definition of P_i as:

$$P_i = L_{ij} : j = 1, J \tag{3}$$

where J is the number of subcomponents or lumps of cross section P_i. Using the notation L_{ij} to denote a lump of a given cross section, a graph or tree can be constructed relating the connectivity of the lumps of the various cross sections. For example, the following tree could relate three cross sections where the middle cross section has two lumps:

$$\begin{tikzcd} & L_{11} \arrow[dl] \arrow[dr] & \\ L_{21} & & L_{22} \arrow[d] \\ & & L_{31} \end{tikzcd} \tag{4}$$

Next, nodes within the tree are categorized. For each tree node, its surface area is calculated and then compared to child and parent nodes to determine if the current node is a local maxima or local minima in cross sectional area. Local minima in particular are important as they typically indicate where one deposition layer of material joins a material deposited at a later time. If an extrema in cross sectional area occurs at a head or terminal node then special process masks may be required. No masking decisions are made at this stage; rather these nodes are just marked so that they can receive attention during the mask reconciliation stage.

Once the nodes are categorized, the tree is traversed to find the cross sections required to build the device. It is assumed at this stage that the surface micromachining process proceeds by depositing a layer of material, using a mask to etch away unwanted material, removing the mask and then repeating this process. This is a simplification that real processes do not necessarily follow which can be accounted for at a later stage as covered in the next section. While traversing each branch of the tree, first the locations of local minima nodes are recorded and between any pair of local minima on a given branch, a local maxima is sought. Local maxima nodes are typed as *poly* masks as they typically represent how a structural layer like a polysilicon layer was masked before etching; *poly* is purely a name of convenience as this method would work for any material. Similarly, local minima nodes are typed as *sac-ox* masks as they typically correspond to masks used in etching sacrificial layers like the sacrificial oxide layers, SiO_2. Again, this nomenclature is for convenience. Terminating nodes that end in local extrema are typed as *dimples* if they are local minima or *undercuts* if they are local maxima. These two mask types are almost equivalent to *sac-ox* and *poly* masks respectively, however their use in a fabrication process is different from *sac-ox* and *poly* masks so they are singled out at this stage.

The masks thus far identified have an additional attribute associated with them. Each masks has a *thickness* which corresponds to the difference in height between the node where the mask was identified and either the next extrema on a child branch or the end of the current branch. When attempting to match or reconcile the masks found from the topology analysis with masks required for given process, this thickness is used to determine if a given process step is compatible with a given mask. Next, these masks will be converted to production masks.

2.2 Creating Masks

The candidate masks found in the previous section apply only to an idealized version of surface micromachining as was assumed earlier. If one were only given a model of a part, and the part's designer did not have a specific production process in mind for that part, then the candidate masks together with their thicknesses and material types would define a new, idealized production scheme for this device.

However, if the designer of this part had a specific production process in mind then the candidate masks must be reconciled with process mask specifications to yield valid masks as follows. First the process specification is searched for the materials and material thicknesses it uses, masks names and their locations in the process stream. Next, the target process is searched for places where the assumed deposition-mask-etch process order does not occur. With these parameters known, the candidate masks can be searched for masks that match the function of those used in a given process step. If a candidate mask corresponds to a layer which is thicker than layers in the target process, then that mask can be duplicated and used to produce two laminated layers in the actual process. If all of the candidate masks cannot be fit to the target process then the designer can be informed of what feature is blocking this fit.

3 METHOD

The analysis described in the previous section forms the basis of the following algorithm, which successfully infers 2D mask sets from complex 3D models. Aspects of the algorithm that have not yet been discussed concern largely logistical points. For example, a given 3D model will have many non-intersecting bodies. It is efficient to work on one body at a time, so initially the model is divided into its non-intersecting components. Compensation for this division occurs later when the mask sets are summed. This summation is straightforward as the non-intersecting bodies will have non-overlapping masks. Finally, a simplification of the topology tree is conducted where redundant nodes are joined, a process where by nodes that topologically connect the same nodes are combined to one node. Given a 3D model, the algorithm is:

1. Locate independent bodies.
   (a) find all non-intersecting bodies
   (b) separate bodies made of different materials
   (c) separate bodies only connected via. ground
2. For each body
   (a) Generate a topology tree.
   (b) Categorize the nodes of the tree.
   (c) Combine redundant nodes.
   (d) Locate deposition boundaries.
3. For each deposition domain
   (a) Locate masks
   (b) Save masks in candidate mask set
4. Sum all candidate masks
5. Reconcile masks with the target process.

It is significant to note that specific process details do not enter the algorithm until the final step. Allowing most of the algorithm to operate independently of process details keeps the algorithm flexible to process changes.

3.1 Implementation

The algorithm was implemented in a C++ program called *Faethm* , and figure 1 illustrates its overall integration in the design flow. Typically, a 3D model is generated in a 3D design environment and passed off to *Faethm* . *Faethm* load the model and reduces it to cross sections in libraries specific to the vendor of that model format. Next, the cross sections are transfered to a custom 2D modeling library which has been optimized for such objects. This allows a minimum subset of the code to be tailored for specific 3D model formats while producing masks in a format consistent for our design tools. For this work, models were both manually generated and provided by Sandias SUMMiT V 3D Modeler [4]. Manually generated models can be created in AutoCAD with support for ProEngineer currently under development.

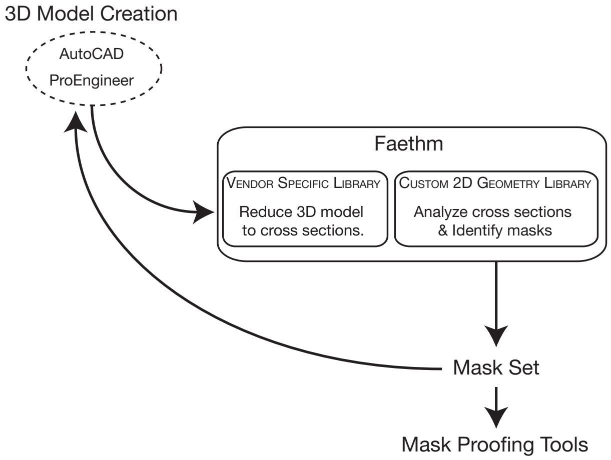
3D Model Creation
AutoCAD
ProEngineer
Faethm
Vendor Specific Library
Reduce 3D model to cross sections.
Custom 2D Geometry Library
Analyze cross sections & Identify masks
Mask Set
Mask Proofing Tools

Figure 1: Implementation and integration of *Faethm* in a design environment.

3.2 Example

As an illustrative example of this method, figure 2 depicts a hub which is used to hold a gear in place. A single hub is an example of an independent, non-intersecting body found in step one of the method listed previously. The hub is cut into horizontal cross sections and the unique cross sections are assembled into a topology tree. Cross sectioning and creating a topology tree typically scale with the number of vertical surface in a model squared. Thus identifying independent model bodies at the start of this method is an important aspect in method throughput.

Continuing, note that the topology tree developed for the hub is branched and non-symmetric as the center post has a different topology than the outer ring of the hub. Figure 3 demonstrates the analysis of the topology tree. After the area of each topology node is calculated, an area versus height graph is created where the vertical lines connect the nodes to indicate topological relationships. Since the hub's topology tree is branched, the branch for the outer part of the hub is drawn with a dashed line. Using the area data and the topological connectivity of the nodes, candidate masks can be selected.

Reconciliation of the masks with process constraints produces a set of production masks for the hub. Operations on masks to convert candidate masks into pro-

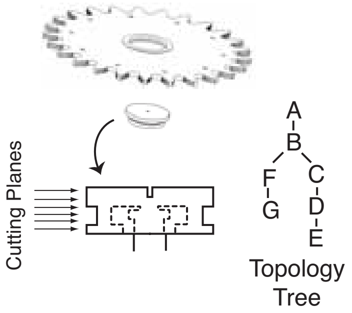

Figure 2: Locating the unique cross sections and building the topology tree for a hub which holds a gear in place.

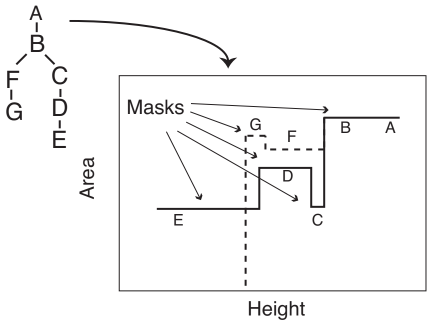

Figure 3: Analyzing the topology tree allows one to locate masks.

cess masks include combining masks that operate on the same material at the same point in a process, discarding redundant mask features and inserting masks required by a given process.

While the part in this example is quite simple, many larger and more complex parts have been used to test the program. On a typical workstation, models containing 1,000 vertices are reduced to masks in under 10 seconds as shown in figure 4. More complex models with 10,000 vertices can be analyzed in approximately 100 to 300 seconds.

4 CONCLUSION

The algorithm presented here and coded in the *Faethm* program is capable of generating accurate mask sets for complex 3D devices. By focusing on a models topology first, this work can identify masks for anisotropic and isotropic (dry and wet) etching processes.

ACKNOWLEDGMENTS

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy, National Nuclear Security Administration under Contract DE-AC04-94AL85000.

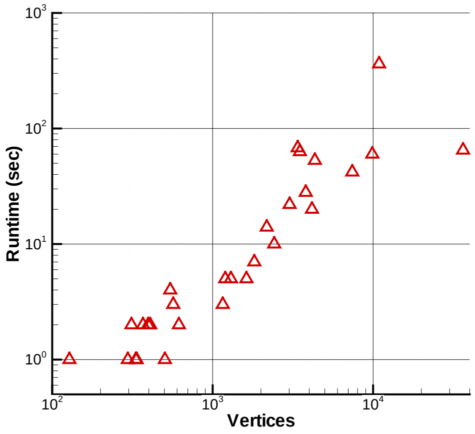

Figure 4: Typical program run times for models of increasing complexity (higher vertex count for larger more complex models).

REFERENCES

[1] L. Ma & E. Antonsson, "Robust mask-layout and process synthesis" *J. of Microelectromechanical Systems* vol. 12, no. 5, pp. 728–729 (2003)

[2] V. Venkataraman, R. Sarma & S. Ananthasuresh, "Part to Art: Basis for a Systematic Geometric Design Tool for Surface Micromachined MEMS" *Proceedings of the ASME Design Engineering Technical Conferences 2000* pp. 1–14 (2000)

[3] S. Cho, K. Lee & T. Kim, "Development of a geometry-based process planning system for surface micromachining" *Int. J. of Production Research* vol. 40, no. 5, pp. 1275–1293 (2002)

[4] C. Jorgensen & V. Yarberry, "A 3D Geometry Modeler for the SUMMiT V MEMS Designer" *Modeling and Simulation of Microsystems 2001* pp. 594–597 (2001)