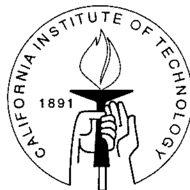
Computer Aided Mask Layout Synthesis for Anisotropic Etch Photolithography

Thesis by
Mark K. Long

In Partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy



California Institute of Technology
Pasadena, California

1999

(Defended May 6, 1999)

© 1999

Mark K. Long

All rights Reserved

Computer Aided Mask Layout Synthesis

for

Anisotropic Etch Photolithography

by

Mark K. Long

In Partial Fulfillment of the

Requirements for the Degree of

Doctor of Philosophy

Abstract

The increasing penetration of MEMS technology into new application domains suggests the need for sophisticated engineering tools that can automate routine MEMS engineering design functions. This thesis discusses the development of algorithms and automated software tools that are intended to automate the mask-layout process for bulk etch micro-machining.

At present, a designer conceives of a MEMS function, then (informally) creates a mask-layout that the designer believes will process into a shape that will exhibit the desired function. Because of the highly anisotropic nature of the bulk etching process, the mask design process relies heavily upon the designer's intuitive understanding of the etching process. A prototype device is created from the candidate mask, and its actual function is tested. This process can result in many iterations, and many prototypes.

This dissertation presents a method to automatically synthesize the mask layout for a bulk etching process. That is, given a desired part geometry and process parameters, the algorithm determines a candidate mask geometry that will etch to the final desired shape even in the case of highly anisotropic etchants. It will also compute compensation structures for difficult to etch features. Conceptually, the algorithm is based on the use of a forward etch simulation in reverse time.

Since the forward etch process is a many-to-one map, the reverse time simulation is augmented to include a choice of valid preimages. Timing models are introduced to devlop mask layouts that have appearing crystal planes during the etch and shown to allow more complex compenstation structures.

Finally, calibration masks are developed that allow the experimental determination of etch rate parameters without painstaking measurements.

Contents

1 Introduction **1**

1.1 Current MEMS-CAD Research 3

1.2 Bulk Etching 5

1.3 Current Practice 5

1.4 Overview of Approach 6

1.5 Contributions and Organization of this Dissertation 7

2 Background **9**

2.1 Photolithography 10

2.2 Etching 13

2.2.1 Isotropic Etching 13

2.2.2 Anisotropic Etching 13

2.3 Etch Rate Modeling 16

2.3.1 Higher Order Etching Effects 16

2.4 Mask Layout and Etch Simulation 17

2.4.1 Simulation 18

2.4.2 Special Purpose Masks 21

2.5 Summary 22

3 Approach/Methodology **23**

3.1 Geometric Etch Modeling 25

3.1.1 Geometric Etch Computation 31

3.2 Mask Layout Approach 35

3.3 Summary 42

4 Mask Synthesis **43**

4.1 Simple Concave Vertices 44

4.1.1 Simple Concave Direct Preimage 44

4.1.2 Simple-Concave Inner-Hull Indirect Preimage 46

4.1.3 Alternative Indirect Simple-Concave Preimages 49

4.2 Simple Convex 51

4.2.1 Simple Convex Direct Preimage 51

4.2.2 Simple Convex Outer-Hull Preimage 52

4.3 Compound Convex 54

4.3.1 Three Vertex Simple Compensation Structures 56

4.3.2 Four Vertex Simple Compensation Structures 59

4.4 Compound Concave 60

4.4.1 Compound Concave Approximate Preimages 61

4.5 Example 64

4.5.1 Compound Convex Vertices 66

4.5.2 Compound Concave Vertex 67

4.5.3 Simple Vertices 69

4.6 Robustness 71

4.7 Summary 75

5 Timing Models for Mask Synthesis **76**

5.1 Compensation Structure Example 79

5.1.1 SIMULATION 86

6 Calibration **89**

6.1 ⟨110⟩ Etch Rate and Orientation 90

6.2 Fast Plane Etch Rate and Orientation 93

6.3 Intermediate Plane 97

6.4 Summary 97

7 Algorithm Design and Implementation **99**

7.1 Polygon Layer 100

7.2 Vertex Classification 101

7.3 Preimage Calculation Algorithm 102

7.3.1 Indirect Preimage Algorithm 102

7.3.2 Direct Preimage Algorithm 105

7.3.3 Timing Model 105

7.4 Design and Analysis Plotting Algorithms 105

7.4.1 $\phi\alpha$ Diagram 105

7.4.2 $\gamma-$ Velocity Plot 107

7.4.3 $\beta_2\beta_3$ Diagram 107

7.5 User Interface and High Level Structure 108

7.5.1 Client-Server Architecture 109

7.5.2 User Interface and Operation 111

7.6 Summary 111

8 Conclusions and Future Work **112**

8.1 Areas for Future Development 113

8.1.1 Intelligent Geometry Modification 113

8.1.2 Improved Calibration 114

8.1.3 Multi-Polygon Mask Layouts 116

8.1.4 Three-Dimensional Micro-Structures 116

List of Figures

1.1 Design and Fabrication Process. 2
1.2 Polar Etch Rate Diagrams (etch rate R vs. crystal orientation θ: 2D-Anisotropic, 3D-Anisotropic [23]). 6

2.1 Wet etching sequence. 12
2.2 Isotropic etching (A) with agitation (B) without (diffusion limited). 14
2.3 Sample mask shapes. 15
2.4 Anisotropic etch results, appearing and disappearing planes [23]. . . 15
2.5 Etch rate and etch reluctance diagram for wet etching silicon. 19
2.6 Wulff-Jaccodine construction [19]. 20

3.1 Polar etch rate diagrams 26
3.2 Frank's reluctance diagram and etch evolution [18]. 27
3.3 Etch rate and etch reluctance (slowness) diagram for KOH on silicon. 28
3.4 Convex hull construction over slowness (reluctance) diagram, after [51]. 29
3.5 Convex hull construction over slowness (reluctance) diagram. 30
3.6 Vertex velocity computations diagram. 32
3.7 Vertex velocity example. 35
3.8 "Reverse simulation" of etching. 37
3.9 Appearing planes during etch. 37
3.10 Example of topology modification and reverse simulation. 39
3.11 Example of preimages for compound convex vertices, i.e., compensation structures. 40
3.12 Structure of simple concave solution approach. 41

4.1 (A) Concave vertex, (B) Etch rate diagram, (C) Reluctance (Slowness) diagram. 45
4.2 Convex hull construction over reluctance diagram. 46
4.3 Direct preimage construction. 46
4.4 Indirect preimage construction with comparison to direct preimage. . 48
4.5 Topology modification during indirect preimage construction. 48
4.6 Alternative indirect preimage construction. 50
4.7 γ-Velocity Plot. 50
4.8 Alternative indirect preimage construction. 51
4.9 (A) Convex vertex, (B) Etch rate diagram, (C) Reluctance (Slowness) diagram. 52
4.10 Convex vertex overlaid with reluctance diagram. 52
4.11 Direct preimage construction. 53
4.12 Indirect preimage construction. 53
4.13 (A) Compound convex vertex, (B) Etch rate diagram, (C) Reluctance diagram. 55
4.14 Inner and outer hull construction. 55
4.15 Parameterized generic vertex and three point compensation structure. 57
4.16 $\phi\alpha$ Plot, dark areas are simple vertices. 57
4.17 $\beta_2\beta_3$ plot, colorbar indicates distance from A to A'_1. 58
4.18 Resulting compensation structure. 58
4.19 Simple four vertex compensation structure parameterizations. 59
4.20 $\beta_2\beta_4$ diagrams for four simple vertex compensation structures. Dark regions represent valid structures. 60
4.21 (A) Compound concave vertex, (B) Etch rate diagram, (C) Reluctance diagram. 61
4.22 Construction of approximation for compound concave vertex. 62
4.23 Compound concave vertex with improved approximation. 63
4.24 Desired shape for mask synthesis example. 65
4.25 Etch rate and etch reluctance diagrams. 66

4.26 Compound convex vertices decomposed into multiple simple vertices. 67

4.27 Compound concave vertex with approximation. 68

4.28 Final mask layout (solid line) with original shape (dashed line). . . . 71

4.29 Simulation results for example mask layout. Outer shape is the computed mask, equal etch time steps to final etched shape. 72

4.30 Experimental validation of corner compensation [64]. 73

4.31 $\beta_2\beta_3$ diagram, greyscale refers to maximum vertex velocity. 74

4.32 Under- and over- etched compensation strucutres. 74

4.33 Under-etched simple convex preimages. 75

5.1 Vertex merging during preimage construction. 77

5.2 Preimage construction with timing model. 80

5.3 Preimage construction with timing model. 81

5.4 Compensation structure for vertex A, after [65]. 82

5.5 Basic 3 point compensation structure. 82

5.6 Insert $\langle 100 \rangle$ planes at time, $-t_1$. 83

5.7 Insert $\langle 310 \rangle$ planes at time, $-t_2$. 84

5.8 Insert $\langle 110 \rangle$ planes at time, $-t_3$. 85

5.9 Simulation results of computed compensation structure. 87

5.10 Experimental results of compensation structure [65]. 88

6.1 Mask layout for $\langle 110 \rangle$ orientation determination. 91

6.2 Alignment results etched in $\langle 100 \rangle$ silicon, from [15]. 91

6.3 Expanded mask pattern for rate determination. 92

6.4 Simulation of $\langle 110 \rangle$ rate calibration pattern with 5° misalignment. . 93

6.5 Simulation of fast plane rate calibration pattern. 94

6.6 Generalized vertex diagram $(\phi - \alpha)$. 95

6.7 Calibration pattern for fast plane orientation determination. 95

6.8 Appearance of new planes on misaligned mask. 96

6.9 Results of Simulation of Calibration Mask. 96

6.10 Mask and simulation results for $\langle 010 \rangle$ rate calibration. 98

7.1 Algorithm for decomposing vertices of each polygon. 101
7.2 Algorithm for convex hull calculation. 103
7.3 Algorithm for the vertex classes. 104
7.4 Algorithm for direct inverse and timing model final calculation. . . . 106
7.5 CAMLE Operation. 110

8.1 $\phi\alpha$ Diagram Shows Classification of a Generalized Vertex. 114
8.2 $\phi - \alpha$ Diagram Augmented with Appearing Plane Rate Information. 115
8.3 Part Geometry Overlayed on $\phi - \alpha$ Diagram. 115
8.4 Mask layout for improved compound concave vertex. 117
8.5 Surface Edges and Appearing Sidewalls. 118
8.6 Appearing Sidewall Angle vs. Surface Orientation. 119

Chapter 1

Introduction

The increasing penetration of micro-electro-mechanical-systems (MEMS) technology into new application domains suggests the need for sophisticated engineering tools that can automate routine MEMS engineering design functions. Such tools will allow non-specialists to design and deploy reasonably sophisticated MEMS devices and will enhance the efficiency of expert designers. This thesis discusses the development of algorithms that automate the mask-layout process for bulk etch micromachining. Bulk etching is a mainstay in the arsenal of fabrication technologies used by engineers designing crystalline structures in MEMS and other applications. A comprehensive suite of design and manufacturing (CAD/CAM) tools will accelerate the development and deployment of MEMS devices.

While there has been a variety of work in the general area of computer aided engineering (CAE) for MEMS, many of these tools apply to process simulation or analysis of the functional or mechanical properties of a device; an extensive review will be given in the next section. Often manufacturing issues are not addressed. See Figure 1.1. This work is complimentary to these prior developments in that the effect of fabrication issues on the design is being addressed. Thus, this work has an analogous role to that of traditional macro scale computer-aided-machining (CAM).

Traditional computer aided (macro scale) machining has two basic parts. Computer aided machining (CAM) software packages are used to set up the part geometry and generate the method to machine the part. This research considers how some

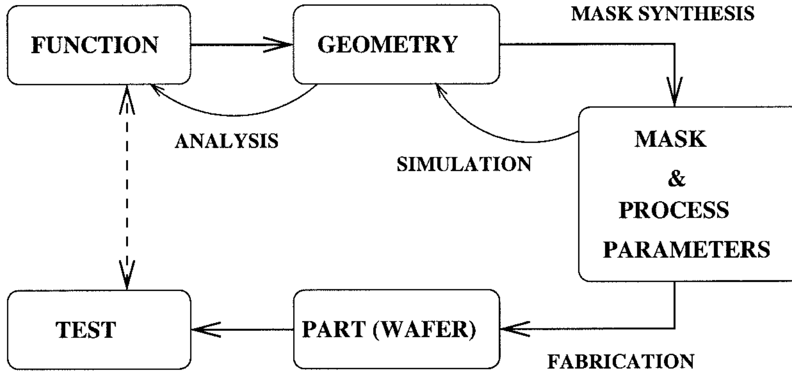


Figure 1.1: Design and Fabrication Process.

similar benefits may be provided in the micro-machining process. The second part is the actual physical machining of the part which is handled by a CNC (computer numerically controlled) machine tool. In micro-machining this latter component is done with bulk chemical milling (etching) processes.

While some attributes of mechanical parts are important regardless of scale, there are fundamental differences in the machining process for macro and micro machined parts. Therefore, few of the CAM tools transfer directly. However, functional analogies can be drawn between macro CAM and micro-scale mask-layout. The capabilities of traditional macro-scale CAM tools suggests the following goals for a computer aided machining (μCAM) tool:

- Automate the mask synthesis procedure.
- Aid in selecting the proper etchant and process parameters (temperature, concentrations, etc.).
- Reduce design iterations.
- Increase quality, tolerance, finish, and repeatability.
- Shift focus to part functionality from fabrication procedure.

1.1 Current MEMS-CAD Research

Developments in computer aided engineering for MEMS have been recognized as important to the development of the field. Current research in the field is investigating a number of tools for MEMS design. Figure 1.1 shows a design and fabrication process for a generic device. The top half of the diagram is design and analysis "information" while the bottom half corresponds to prototypes or finished hardware devices. Design analysis and changes in the top half of Figure 1.1 are quick and relatively inexpensive, while device experiments and changes in hardware (on the bottom half of the diagram) are costly and time consuming. This fact has motivated extensive research in the development of analytical techniques and software tools direct at those design phases represented by the top half of Figure 1.1.

Stephen Senturia and his group at MIT have done extensive work in the area of computer aided analysis and simulation of integrated MEMS devices [33, 50, 49]. One of the thrusts of their work has been on mixed mode simulation where electrical and mechanical models are combined, e.g., capacitance calculations for a mechanically deformed diaphragm.

Koppelman at IBM developed a program called OYSTER that simulates the geometric effects of sequential IC process stages [30]. The results of OYSTER can then be used with various analysis packages including finite elements. It was originally developed for IC simulation but is applicable to MEMS systems manufactured with IC processes.

A number of other researchers have applied finite element modeling techniques to MEMS problems. Crary and Zhang [12] developed their *CAEMEMS* software workbench which provides a material properties database, solid modeling, and FEM simulations of components and systems. Pourahmadi and Twerdok [40] focused on sensor material property analysis with FEM. Schwarzenbach et al. [47] developed extensions to their *SESES* system to analyze MEMS devices with FEM.

Mastrangelo and his group at University of Michigan have developed *MISTIC*, a process compiler for thin film microdevices [22]. *MISTIC* employs a systematic method for the automatic generation of fabrication processes of thin film micro-

machined devices.

Baltes et al. at the Physical Electronics Laboratory at ETH Zurich have developed ICMAT and SOLIDIS [3]. ICMAT is a database obtained from measuring process dependent CMOS thin film electrical, magnetic, thermal, and mechanical properties by using dedicated material characterization micro-structures. The toolbox SOLIDIS provides coupled numerical modeling of electrical, magnetic, thermal, and mechanical phenomena.

Sethian has adapted the mathematical formalism of level sets to model etching deposition and lithography [2, 52]. This work focuses on computational issues in a broad range of surface evolution problems in microfabrication.

A number of groups have developed simulations systems for anisotropic etching. Hubbard and Antonsson developed etch simulators based on both eigen-shape and cellular automata approaches which allows the user to enter etching model properties and a mask layout and result in simulated result of the anisotropic etch [25, 24, 23]. Buser and de Rooij developed *ASEP* [9, 7], a computer program for which simulates single crystal etching in KOH. Sequin developed the slowness method [51] which will be discussed in more detail in § 3.1. Toh, Neureuther, and Schekler [58] have developed a three-dimensional etch simulation based on a novel ray-string algorithm that provided improved computational efficiency. Danel and Delapierre [13] developed an etch simulation program based on extensions to the Wulff-Jacodine [19] prediction method.

Recently, Li and Antonsson at Caltech have developed methods for synthesizing mask layouts for bulk etching using evolutionary algorithms wrapped around an etch simulation [31].

The review of recent research in MEMS-CAD in this section shows that the vast majority of current research on design tools for MEMS has focused on the analysis of the function of MEMS devices or the simulation of fabrication processes. Both of these general areas are very important in the development of MEMS devices but are incomplete in their coverage. One of the areas of investigation that has been largely ignored has been automated design tools to develop fabrication process

specifications from desired device geometry. This is precisely the focus of this thesis and is complementary to this previous research. The work in this thesis on mask synthesis is complementary to this previous work and is put in context by Figure 1.1.

1.2 Bulk Etching

Bulk etching (also referred to as wet or anisotropic etching) is a type of chemical crystal dissolution commonly used in the fabrication of silicon micro devices. It's unique feature is that the rate of crystal dissolution is directionally dependent (a function of the orientation of the etched surface). It is not uncommon for an etchant to have an orientational variance in dissolution rates in excess of 100:1. Figure 1.2-A shows the behavior of a typical etchant. The radius of the plot is the crystal dissolution rate (or etch rate) in microns per hour (μ/hr), and the θ plot coordinate is the orientation of the crystal plane being etched (generated by Matlab with etch rate data). Figure 1.2-B depicts a three-dimensional etch rate diagram for a typical etchant [23].

This work is not restricted to mono-crystalline silicon (Si) or planar wafers; however, the case of Si wafers is highlighted due to its common use.

A mask or stencil is layed on the wafer to control which regions are exposed to the etching solution. The exposed areas of the wafer are chemically milled at a rate given by the orientation of the exposed crystal plane and Figure 1.2. A complete description of this lithography process is given in § 2.1.

1.3 Current Practice

One significant difficulty in the bulk etch fabrication process described above is generating the proper mask-layout geometry such that the result of the fabrication process is the part geometry as the designer had intended.

At present, a designer conceives of a MEMS function, then (informally and intuitively) creates a mask-layout that the designer believes will etch into a shape that will exhibit the desired function. This process is based largely on the engineers

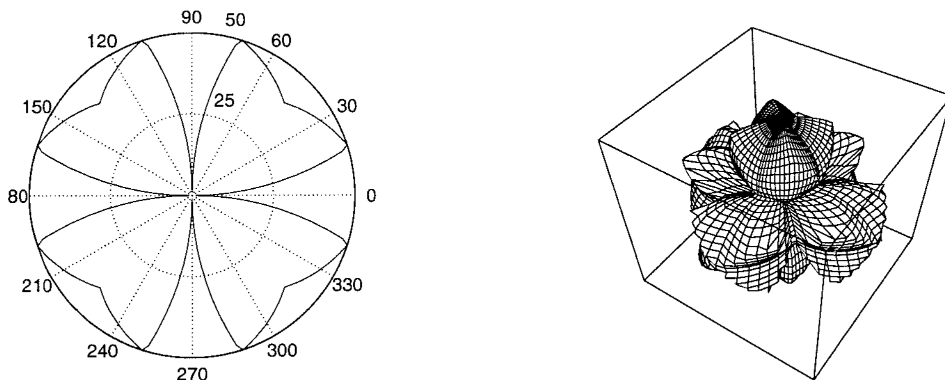


Figure 1.2: Polar Etch Rate Diagrams (etch rate R vs. crystal orientation θ: 2D-Anisotropic, 3D-Anisotropic [23]).

intuition and experience, and is very difficult for complex geometries or for novice engineers. A prototype device is created from the candidate mask, and its actual function is tested. This process may result in many design iterations, and hence many prototypes. In many cases, manufacturing issues dominate the design cycle.

Recently, researchers have developed algorithms to simulate the etching process [26, 8, 13, 23, 56]. While these simulators help reduce the number of prototype fabrication iterations, the mask geometry synthesis task still relies heavily upon the designer's intuitive understanding of the etching process. Additionally, the wide variety of micro-machining processes in use today make this intuitive work even more difficult, and typically force designers to limit themselves to familiar processes. In some cases certain etchants or geometries are avoided because they require too many iterations to perfect the mask-layout.

1.4 Overview of Approach

This dissertation presents a method to synthesize the mask layout for a given etching process. That is, given a desired part geometry and process characteristics, the

method determines a candidate mask geometry that will etch to the final desired shape, even in the case of highly anisotropic etchants. In cases where there does not exist a mask geometry that will yield the desired shape, the approach produces an approximation that can be produced with the specified process.

The mask synthesis procedure can be roughly thought of as a reverse simulation process. The evolution of the part shape during the etch process is a many-to-one map. That is, in many cases there are multiple masks that will etch the same physical geometry. In highly anisotropic etching, crystal planes may appear or disappear during the etching. These changes in the polygonal topology result in the forward simulation being indeterminant when strictly "run backward" in time. The mask layout computed using an inverse simulation that neglects these topology changes is called a *direct preimage.*

By characterizing the appearance and disappearance of crystal planes during etching, the method allows manual or automatic determination of each of these bifurcations. Thus additional layout criteria may be used to direct the inversion process and arrive at alternative solutions. Alternatives computed using the inverse bifurcations are called *indirect preimages.*

In cases where no direct preimages exist, families of valid preimages are computed with parameterizations that allow for selection. In other cases no direct or indirect preimages exist. Here, approximate mask layouts are computed.

1.5 Contributions and Organization of this Dissertation

The main contribution of this dissertation is the first algorithmic method to generate mask-layout geometry for anisotropic etch fabrication. In the course of that development, insight into the mask layout problem and design tools have been developed that contribute to the understanding of the sensitivity and calibration of the bulk etch fabrication process.

The approach provides multiple solutions in some cases and allows for inclusion of additional design criteria in the selection of mask layouts. These methods will

quantify previously intuitive design cycles and allow engineers to use a wider variety of processes and operating regimes. This work will play a role in shortening the time and cost of MEMS design implementation.

While our examples focus on application in the MEMS field, it is equally applicable to fabrication of other mono-crystalline micro-structures such as germanium, quartz, etc.

Chapter 2 presents background information on the etching process and previous work on etch modeling and simulation. Etch rate diagrams (ERD) and higher order effects not described by ERDs are discussed.

Chapter 3 contains a discussion of the philosophical approach and methodology of the synthesis algorithms. Current models of etching are discussed and additional insight is extracted that can be leveraged for synthesis. Data plotting methods that provide qualitative and algorithmic support for the mask-layout engineering task are described.

Chapter 4 discusses methods of mask-layout synthesis in detail. A taxonomy of vertices is described, and procedures for synthesizing each type are presented. The presence of multiple solutions is discussed and approaches for approximation when no exact solution exists is described. The synthesis method developed in this chapter is then applied to an example with each of the classes of vertices.

Chapter 5 presents an extension of the algorithms in Chapter 4 to compute preimages with compound vertices. While direct experimental results were not available to validate the results, the method was shown to duplicate the experimental results of others and to agree with simulation results.

Chapter 6 discusses procedures to experimentally derive etch rate data. Masks are developed that minimize the requirement for tedious quantitative measurement.

Chapter 7 presents the structure of the algorithms and implementation details.

Finally, Chapter 8 summarizes the results in this dissertation and provides details of potentially fruitful future work.

Chapter 2

Background

In the fifteenth century, armor was decorated using a patterned wax mask and acid based etchants. This is one of the earliest known uses of *wet etching* [61, 32]. Wet etching, also called chemical milling, is a material removal process where a material is placed in a liquid bath that chemically dissolves the surface of the material. As in the case of armor decoration, often portions of the surface are covered or "masked-off" to selectively remove material for certain regions. One of the limitations in this work was the accuracy of the masks.

In 1796, Aloys Senefelder found that when Bavarian limestone was inked and treated in a specific way, he could transfer a carved image onto paper. The word *lithography* comes from Latin (which came from the Greek), lithos, meaning stone, and -graphia, meaning "to write." As a result of the introduction of photosensitive masks by Niépce in 1822, the tolerances on the production of masks was greatly improved. The earliest example of the combination of lithography with photosensitive masks (photo-lithography) and wet etching is the 1827 copy of an engraving of Cardinal d'Amboise by Parisian engraver Lamaître in which he demonstrated an accuracy of 1mm [61, 32]. By 1961 photolithographic accuracy had improved to $5\mu m$.

During the 1940s lithography based chemical milling was used in the manufacture of printed circuit boards, followed by use in the manufacture of precision parts in television shadow masks, IC lead frames, and encoder disks in the 50s and 60s. As

early as the 1950's wet etching was used in silicon semiconductor processing [42, 43, 45, 46, 62, 21, 60], using a combination of hydrofluoric acid (HF) and hydro-nitric acid (HNO_3). These chemicals results in an isotropic etch, i.e., the etch rate is independent of the crystal orientation.

Isolation requirements in silicon integrated circuits motived work on high aspect ratio silicon etching. The use of potassium-hydroxide (KOH), water, and alcohol mixtures resulted in highly anisotropic wet etching of silicon, i.e., the etching rate was directionally dependent. The work of the early 60's on isolating integrated circuits with dielectric structures continued through the 1970's when chemical and electro-chemical anisotropic etching was applied to V- and U-groove transistors.

Application of silicon (Si), germanium (Ge), and wet etching to micro-mechanical applications evolved around the same time period. In 1961 Pfann [39] suggested the fabrication of Si piezoresistive sensors for stress, strain, and pressure and Tufte used etching and oxidation processes in the fabrication of thin Si diaphragms for pressure sensors [59]. By 1974 National Semiconductor marketed a variety of mature Si pressure sensors [32], other micro-structure devices followed: Texas Instrument's print heads, Western Electric's fiberoptic alignment structures, IBM's ink jet nozzles [32].

The next phase of the application of silicon as a mechanical material resulted from Peterson's 1982 work [38]. While previous efforts were primarily centered in industry, new research at universities and government labs grew at a rapid pace.

The two key technologies that make wet etching popular are lithography and etching.

2.1 Photolithography

Wet etch fabrication, like other micro fabrication processes, starts with photolithography. An in-depth discussion [32] of the many approaches and materials used in photolithography is beyond the scope of this thesis, but a general overview of the basic technique as used in wet etching follows.

Since a material that is both photosensitive and resistant to common silicon

etchants is not available, a two layer patterning technique is employed. First a thin layer of oxide is grown on the Si wafer surface by heating it to between 900 and 1100°C in a stream of humidified oxygen. For prolonged etching, high density silicon nitride (Si_3N_4) may be applied using low pressure chemical vapor deposition as a robust alternative to silicon oxide.

A thin layer of an organic polymer is deposited on the oxide surface. This material, called the photoresist, is sensitive to ultraviolet radiation. After coating, the photoresist may contain some solvents and residual stresses. To improve adhesion, the solvents and stresses can be reduced by baking the wafer at 75 to 100°C for 10 minutes. The wafer now has a two layer coating of oxide and photoresist (Figure 2.1). Extensive details of the resist coating processes may be found in references [35, 57, 54, 20, 11, 32].

Using the flat of the wafer (Figure 2.1) for alignment, the photoresist on the wafer is illuminated with a UV lamp through the mask. The exposure system is tightly controlled to deliver radiation with the correct spectral content, intensity, and uniformity over the wafer so that an accurate transfer of the mask image onto the wafer is obtained. At this point the image is called "latent," that is, the exposure of portions of the photoresist has resulted in a chemical change in the photoresist, but no pattern exists until the photoresist is developed.

The latent image on the photoresist is developed into a relief image through batch immersion in an agitated bath of a solvent (developer). Photoresists are categorized as *positive* or *negative* resists. The exposure of a positive photoresist typically weakens the polymer and the exposed part of the resist becomes more soluble in the developer. Conversely, the polymer in a negative resist is strengthened by exposure, and thus the resist not exposed is dissolved relatively faster in the developer [6]. The relief pattern on the wafer is now either the positive or negative image of the illumination mask pattern.

The pattern in the photoresist is used as a mask for the oxide layer. The wafer is placed in a bath of hydrofluoric acid (HF) or HF + NH_4F which dissolves the oxide but not the photoresist. The silicon substrate and the oxide under the photoresist

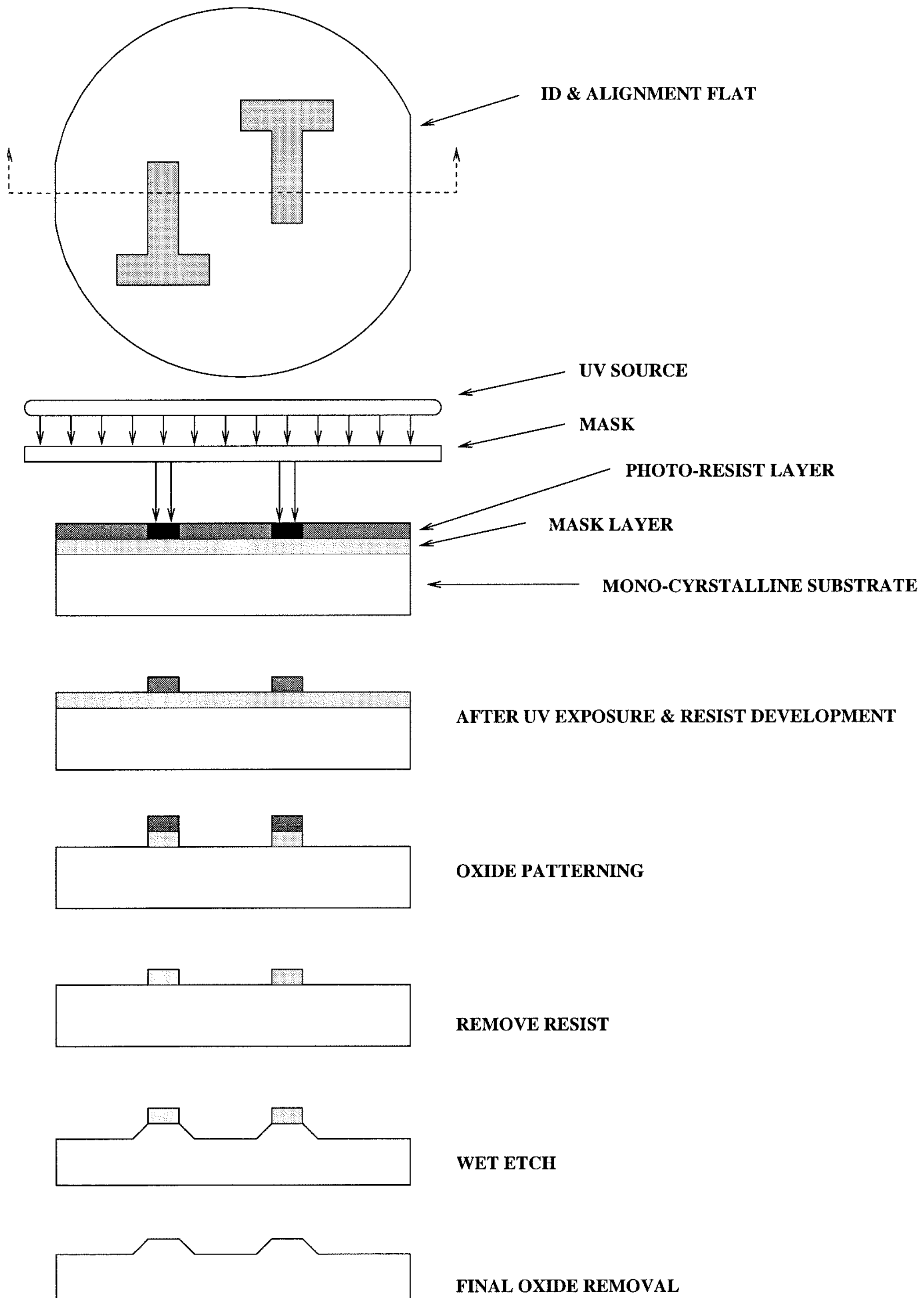


Figure 2.1: Wet etching sequence.

are protected. Now that the oxide has been "patterned" in this last step, the remaining photoresist is removed with a strong acid. Solutions such as H_2SO_4 are used to remove photoresists, but not dissolve any of the oxide or the silicon substrate. What remains is a Si wafer with a patterned oxide that is used as a mask for the wet etch.

2.2 Etching

Wet etching is a process in which exposed surfaces of a material are removed by a chemical reaction. First the reactants are transported to the material surface, the chemical reaction takes place, and the reaction products are transported away from the reaction cite and replaced by new reactants. If the etching rate is determined by the transport mechanism, it is diffusion limited and can be increased by stirring. Otherwise it is reaction rate limited and depends strongly on temperature, etching material, and solution composition. In general, reaction rate limited etching is more repeatable than diffusion limited etching.

2.2.1 Isotropic Etching

Isotropic etchants, by definition, etch in all crystallographic planes with a uniform rate. Thus, the resulting geometry in the silicon substrate has rounded corners. The most commonly used etchants are mixtures of nitric acid (HNO_3) and hydrofluoric acid (HF) [42]. Isotropic etchants can show diffusion limitation and therefore process parameters including agitation can effect the results. Since precisely controlling agitation in a bulk process can be difficult, isotropic etching provides less lateral and vertical control than anisotropic etching [27], Figure 2.2.

2.2.2 Anisotropic Etching

Silicon wafers are manufactured from ingots of single crystal silicon. The periodic positioning of atoms in the crystal is called the lattice. The enumeration of the planes of the lattice are called Miller indices [32]. Each set of Miller indices cor-

Figure 2.2: Isotropic etching (A) with agitation (B) without (diffusion limited).

responds to a different cross section of the crystal. Cross sections have individual atomic layout patterns and packing densities. Silicon wafers are identified by the Miller index of the wafer surface. The $\langle 100 \rangle$ wafer orientation is the most common, although $\langle 110 \rangle$ and $\langle 111 \rangle$ wafers are used in specialized applications. Each wafer also has an orientation flat, Figure 2.1. Typically the flat on a $\langle 100 \rangle$ wafer is parallel to the $\langle 111 \rangle$ direction and is accurate to 3°.

The lattice structure results in etching rates that vary according to the Miller index of the exposed surface. The etch rates can vary from 50:1 up to 200:1 for highly anisotropic etchants. That is, while an exposed $\langle 111 \rangle$ plane may etch at 1 μm/hr, an exposed $\langle 100 \rangle$ plane on the same wafer may etch at 80 μm/hr. The anisotropic etch results in shapes which are bounded by perfectly defined crystallographic planes. The dominance of certain crystallographic planes and reaction rate limitation (relatively insensitive to agitation and diffusion limiting parameters) make the anisotropic etch relatively repeatable; however, it is nonetheless sensitive to temperature and solution composition. Also since the etch proceeds along crystallographic planes, the final geometry may not be geometrically similar to the mask pattern. Dominant crystallographic planes not in the mask pattern may appear, replacing portions of the mask geometry.

The masks in Figure 2.3 result in the shapes in Figure 2.4. The top row of shapes in both figures are *pegs*, that is the interior of the shape is masked off from the etchant and the etching proceeds from the exterior inward. The bottom row is the opposite case, *holes* are etched from the interior of the shape outward. Describing the macroscopic behavior of crystal etching, Batterman [5] stated, "On a concave surface, the plane at a relative minimum in etch rate versus orientation will limit,

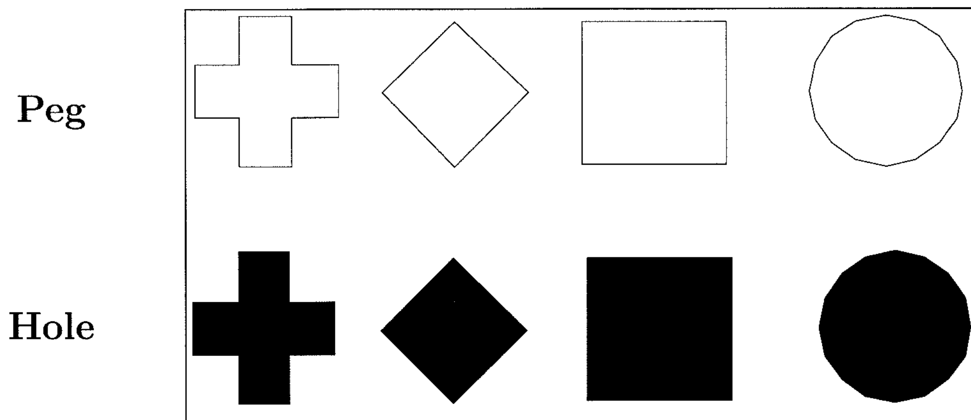


Figure 2.3: Sample mask shapes.

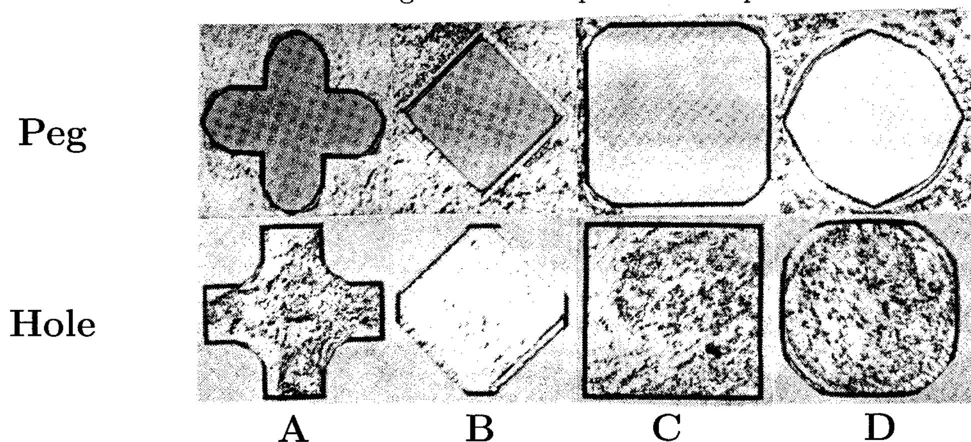


Figure 2.4: Anisotropic etch results, appearing and disappearing planes [23].

while on a convex surface the converse will hold." This is directly reflected in Figure 2.4, Column B; the diamond shaped *hole* (concave corners) has $\langle 110 \rangle$ planes (relative minimum etch rate, Figure 2.5) appearing to limit the etch. In Figure 2.4, Column B, the diamond shaped *peg* (convex corners) has no appearing planes to limit the etch. The second half of Batterman's rule is reflected in Figure 2.4, Column C. The square peg (convex corners) has appearing $\langle 310 \rangle$ planes (relative maximum etch rates).

2.3 Etch Rate Modeling

In order to properly use an anisotropic wet etch process, the etching behavior and anisotropy must be characterized. A variety of models have been proposed ranging from detailed models of the chemical and transport mechanisms to high level geometric models.

Models by Seidel [48] and Elwenspoek [14] describe the anisotropy in terms of the atomic structure of the crystal, bond energy states, and theories from crystal growth. These models tend to be very complex and not very useful for engineering applications. So far, these models have only been developed for a few of the dominant crystal planes; therefore, they do not offer a general description of the wet etching process.

To overcome limitations in the chemical models for etching, several models based on a geometric characterization of etch behavior have been proposed [18, 26]. These models describe either the evolution of crystal planes or plane intersections (vertices) during wet etching, and are calibrated using experimental data. The etch rates for a few of the dominant crystal planes are experimentally derived and geometric interpolation is used to compute the rate for any intermediate lattice cross section. The experimental data and interpolated data are combined to form an *etch rate diagram* (ERD), Figure 1.2. The ERD describes the dissolution rate for a crystal plane based on orientation. It does not, however, indicate the trajectory for the intersection of planes, nor does it directly indicate the elimination or creation of crystal planes.

2.3.1 Higher Order Etching Effects

While the ERD describes the etch rate for individual crystal planes, it does not completely characterize the result of an etch. Variances in surface roughness, hillocks and pitting, pillowing, surface notching (macroscopic roughness) can result from impurities in the etch solution, crystal impurities or dislocations, very high or low etchant concentrations, high temperature etching, etc. Some of these deviations can

be "touched-up" using a slight isotropic etch, while others require careful attention to etchant selection and process parameters.

In Section 2.1, the patterned oxide was described as a mask for wet etching. While it is resistant to etching, it is not immune, and has its own dissolution rate, albeit much lower than Si. For very long etches, pin holes may form in the mask, the mask edges may deteriorate, or the mask may start to separate from the substrate. Each of these conditions may cause deviations in the outcome of the etch from those predicted by the ERD; these are called "secondary" etch effects.

Currently, no comprehensive models exist to describe or predict these "secondary" effects. However, the last 30 years of work in wet etching of Si wafers has resulted in vast accumulated empirical knowledge in the research literature. Careful selection and control of the wet etch process can minimize these effects.

2.4 Mask Layout and Etch Simulation

In order to fabricate some desired geometry using a wet etch on a silicon wafer or other single crystalline material, the process parameters must be selected (etchant, solution composition, temperature, etc.) and a set of masking patterns must be layed out for the photolithography. This "mask layout" must be developed so that the result of the wet etch process has the desired resultant geometry. Since there is no analytical description of this mask layout, and the high anisotropy of the process can make the layout counter-intuitive, researchers have looked for ways to evaluate the mask layout without experimentation. In addition to the time consuming nature of trial and error experimental validation of the mask layout, if the wet etch process is one of many processes in a sequence, the time cost of trial and error validation may be that much higher. Therefore, researchers have used the etching models described above to develop computer simulations of the etching processes to validate mask layout. Since the ultimate goal is mask layout synthesis (ie. reverse simulation), simulation techniques that may be reversible are of special interest.

2.4.1 Simulation

Chemistry based simulators have been developed by Camon [10] and others based on an atomic scale simulation of wet etching that assumes an atomic etching probability depending on the local surface configuration. The simulators are very inefficient and rely on complex algorithms that provide little insight into the global behavior of the etchant. Sangwal [44] has a thorough review of analytic chemical models that attempt to describe the dissolution (etching) behavior of crystals. While various analytic models for isotropic etchants are available, those that attempt to describe the behavior of highly anisotropic etchants are challenged by the formation of sharp edges in the crystal and etch rate profiles which are non-smooth (analytically). To overcome these problems analytic formulations that draw on analogies to shock wave formation are used. While use of shock wave formalisms may work to describe the *forward* etching behavior, a shock is a non-reversible physical event which precludes the use of these models for mask layout synthesis work in this thesis.

In a similar vein, Sethian [2, 52] made an analogy between etching and wave front propagation. This allowed the use of a rigorous analytical formulation using level sets of Hamiltonian systems. Sethian had good success applying this formalism to various types of material deposition and isotropic etching. However, when applied to the highly anisotropic etching considered in this thesis, the formalism had to draw on shock event and entropy theories to form the requisite sharp corners in the crystal. This formulation required numerical schemes that correctly extend the solutions beyond when they are no longer differentiable. These "weak solutions" and approximations have demonstrated some effectiveness for simulating certain types of etching; however, they eliminate the reversibility condition that is required to determine the mask layout from the result of the etch.

Geometric models have also been used to develop computer simulations of the etching process. Many of these simulators can trace their roots back to Batterman and Frank. In 1957, Batterman [5] stated, "On a concave surface, the plane at a relative minimum in etch rate versus orientation will limit, while on a convex surface the converse will hold." The two-dimensional visualization of this remark,

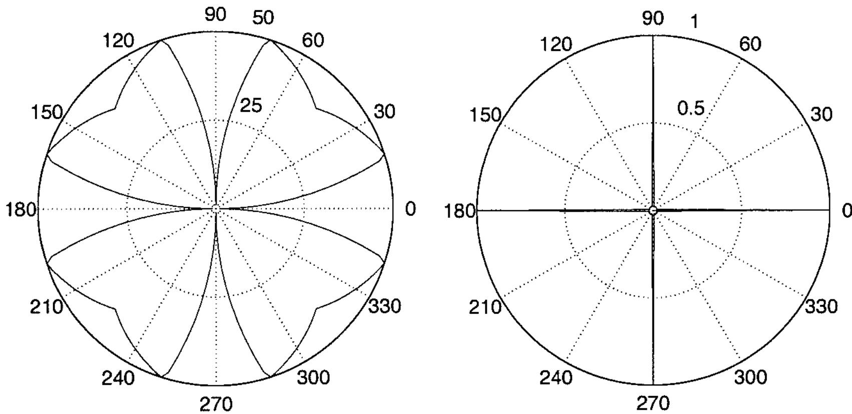


Figure 2.5: Etch rate and etch reluctance diagram for wet etching silicon.

Figure 2.4, indicates that at a concave vertex the lattice plane with the local minimum etch rate will be dominant and limit the elongation of the neighboring planes, and if that plane does not exist, it will appear. Conversely, at a convex vertex, the locally fastest crystal plane will be dominant, and will appear if not already existing. Frank and Ives [17, 18] described a "reluctance plot," which plotted the inverse of the crystal etch rate versus orientation, Figure 2.5. Large values of the 'r' coordinate on this polar plot are more reluctant to dissolution by etching (very low etch rates). The orientation trajectory is then orthogonal to the tangent to the reluctance diagram at the orientation in question, thus the etching shape is constructed [17, 18]. This procedure will be described in detail in the next chapter.

Jaccodine [26] described a different construction, which used an etch-rate diagram (ERD) to compute an envelope of tangents that represented the etch shapes, Figure 2.6. He used the relationships of Batterman to determine if there were appearing planes.

These approaches evolved over the years and the recent surge in MEMS research has seen various algorithmic implementations of etching simulators. Buser and de Rooij [8] developed a program called ASEP (Anisotropic Etching Program) that uses special properties of the KOH etchant to consider only 4 dominant planes. Séquin and Foote [51, 16] developed a simulator based on a "slowness diagram"

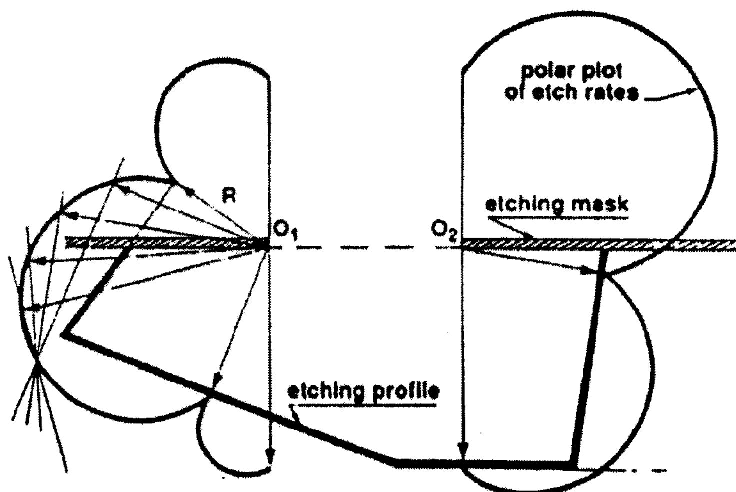


Figure 2.6: Wulff-Jaccodine construction [19].

which is a derivative of Frank's reluctance diagram. Danel and Delapierre [13] improved the Wulff-Jaccodine approach for concave and convex vertices and used the Batterman criteria to handle three-dimensional simulations. Hubbard [23] used an alternative derivative of these models based on *eigen-shapes* to develop a simulation tool. By dividing the geometry up into small segments, self intersection and intersection between features could be computed efficiently.

Another approach to etch simulation has used cellular automata algorithms. Cellular Automata discretizes the volume into many small cells, each with simple rules of interaction, but whose aggregate behavior represents the global result. Than and Bütthenbach [56] used an atomistic model with a stochastic component to simulate three- dimensional wet etching. Hubbard [23, 25] demonstrated that while the coarse global geometric approaches discussed in the last paragraph may be more efficient for simple geometries, in simulations where etch through holes or feature interactions occur, the cellular automata model is very computationally efficient.

Many of these approaches yield good global behavior when compared with experiment; however, few take into account the "higher order" or "secondary" etching

effects discussed in Section § 2.3.1. Thus, even in the case of masks that are validated by simulation, a certain amount of trial and error is still necessary to account for these effects. These models and simulators can reduce design time, but still rely on the engineer's accumulated intuition to layout mask geometry and synthesize new alternatives. This intuitive layout and design process is represented by the forward path in Figure 1.1.

2.4.2 Special Purpose Masks

Because there are no closed form solutions for generating mask layout geometry, researchers have worked to perfect masks for some of the more difficult geometries that arise in mask layouts. The two most difficult are the $\langle 010 \rangle$ or $\langle 001 \rangle$ concave corner and the $\langle 111 \rangle$ convex corner. To repeat Batterman [5], "On a concave surface, the plane at a relative minimum in etch rate versus orientation will limit, while on a convex surface the converse will hold." Thus on a convex surface a plane of relative maximum velocity will dominate. A convex corner formed by the globally minimum $\langle 111 \rangle$ family of planes will be malformed due to the appearance of planes with greater velocity. The solution to this problem has been to create sacrificial volumes of silicon such that under a timed etch, the sacrificial structure will be dissolved by the wet etch, leaving the desired convex corner at the appropriate time. These sacrificial volumes are commonly referred to as *compensation structures* and much research has gone into fine tuning them [1, 4, 28, 34, 37, 41, 63]. The basic approach relies on using multiple vertices in the mask pattern to create a single convex vertex at the completion of the etch process. The dominance of fast planes is balanced with bordering concave vertices whose slow planes dominance results in the desired shape.

The second condition, a concave vertex whose etch is dominated by the locally minimum plane, is more difficult to resolve. Current approaches use sacrificial shapes to delay the dominance of the slow planes, but no mask has been devised to date that can eliminate the appearance of the locally minimum plane. Thus these approaches produce a result that only approximates the desired shape [36].

2.5 Summary

This section has presented the history and development of chemical milling and photolithography with discussion of recent work in the modeling and simulation of anisotropic bulk etching. The geometric etch models used in this thesis are based on the work of Batterman, Frank, and Sequin discussed in this chapter. This thesis extends their previous work in modeling the behavior of crystal etching to the problem of synthesizing lithography that will etch to a desired geometry.

Chapter 3

Approach/Methodology

Traditional computer aided (macro scale) machining has two basic parts. Computer aided machining (CAM) software packages are used to set up the part geometry and generate the method to machine the part. The second part is the actual physical machining of the part which is handled by a computer numerically controlled (CNC) machine tool.

The process of photolithographic masking and wet chemical etching (milling) described in § 2.1- 2.2 is analogous to the CNC machine tool in macro-machining. This work examines the CAM planning part and how some similar benefits may be provided in the micro-machining process.

In addition to geometric modeling of the desired part, traditional computer aided machining provides many improvements to the machining process, such as:

1. Aid in choosing the correct machine tool.
2. Selection or design of specialized end-mills.
3. Decrease in machining time, fixturing, and waste.
4. Increased quality, tolerance, and finish.
5. Reduced need for operator experience and geometric planning.
6. Shift design focus from machining procedure to part geometry.
7. Repeatable robust results.

Over the long history of machining, these and other benefits have been realized through the application of computer aids to the machining process. Compared to macro machining, micro-machining is a very young field. While there has been significant work done in the manufacture of electrical circuits, the machining of micro-mechanical parts have unique issues and requirements. The development of computer aids will assist the maturation of micro-machining.

The above list can be applied directly to computer aided micro-machining (μCAM) with items 1 and 2 replaced with the selection of etchants, process parameters, and crystalline materials.

1. Aid in choosing between available etchants.
2. Selection of temperature, concentration, and additives.
3. Decrease in hillocks, scalloping, and other surface defects.
4. Reduced need for designer experience and intuition.
5. Shift design focus from machining procedure to part geometry.
6. Repeatable robust results.

Currently mask layout is a trial and error process guided by the experience of the design engineer. Their intuition of the process behavior is used to estimate the geometry they think will etch to the correct shape. A combination of etching simulators and test wafer fabrication are iteratively used to converge on a mask layout that produces the desired results.

By directly calculating the size and placement of the features of the photolithographic mask layout, the μCAM method may significantly reduce the number of iterations required. Undoubtedly, test runs will still be required to compensate for unmodeled effects. However, it is hoped that the number will be significantly less when automated mask layout tools are used.

In addition to minimizing trial and error cycles, μCAM tools will allow the MEMS design process to focus on improving the function of the part geometry.

Currently, manufacturing issues dominate the design cycle. This improvement will result from alternative mask solutions being available and also the ability to compare changes in the process parameters and etchant concentrations. Also, some of the ideas developed in the next chapter give an indication of robustness to variations. That is, if the process parameters deviate slightly from their specified values, or the mask is misaligned, will the etchant enter a different regime of behavior?

This identification of distinct regions of behavior, or taxonomy of geometrical features, also allows test patterns to be developed that help calibrate the wet etching process. Due to the scarcity of process calibration data, many facilities constrain the fabrication to a single etchant, solution, and temperature. This availability of test pattern results will greatly enhance the space of available solutions.

While there are a number of specialized mask shapes that have been published in the literature, they only apply to unique geometries, etchants, and alignments. Little indication is provided to allow modification of the pattern for a different orientation or process parameters. By decoding why each of these specialized patterns works, the methods presented here can be used to extend the solution to a wider variety of problems.

In order to accomplish these goals, a model of the material removal process is required. The next section reviews a "geometric" model of the etch process that is the basis of the subsequent work.

3.1 Geometric Etch Modeling

An overview of etch modeling and simulation was given in § 2.4.1. One approach to the mask synthesis problem would be to develop a rigorous analytical formulation of the etching process in such a way that the equations are reversible. As was discussed § 2.4.1, all of the analytical formulations to date have relied on some form of wavefront shock phenomena or entropy considerations in order to properly form the non-smooth surfaces common in anisotropic etching. These analytic formulations result in equations that are non-reversible and this cannot be directly

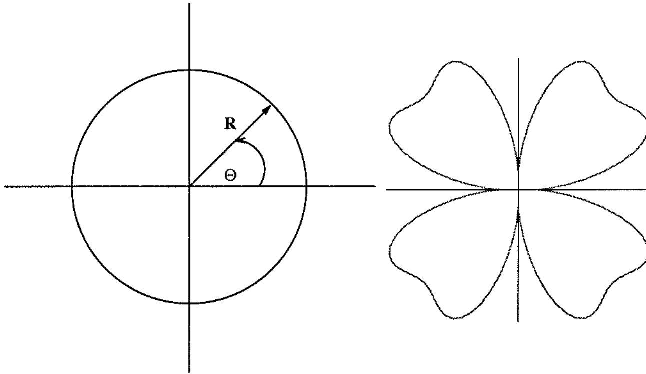


Figure 3.1: Polar etch rate diagrams (etch rate R vs. crystal orientation θ: 2D-isotropic, 2D-anisotropic)

applied to compute mask layouts. Therefore, a geometric model that lends itself to algorithmic reversibility was chosen. This section gives a detailed description of the geometric model employed in this work.

The behavior of a given etchant, process parameters and crystal are encapsulated in the etch rate diagram (ERD) [29, 48, 23], Figure 3.1. The etch rate diagram for a particular process and set of conditions is a polar or spherical plot of the angle of the etch face normal plotted against the rate of etch per unit time in that normal direction. Typically, etch rates for the dominant crystal planes are experimentally determined, and a variety of interpolation schemes are used to fill in the diagram. While the ERD describes the local etching behavior of a crystal plane, it does not indicate how the gross topology of features will change with time. There are a variety of approaches to model this behavior (§ 2.4.1). The model used here has evolved from the work of Batterman [5], Frank [17, 18], and Sequin [51].

Unlike isotropic etching, anisotropic wet etching preserves the geometry/shape of the mask boundary that is being etched with the only changes in shape occurring due to Batterman's rule: "On a concave surface, the plane at a relative minimum in etch rate versus orientation will limit, while on a convex surface the converse will hold." Thus the orientation of crystal planes being etched will be preserved, and the general shape will only change at a vertex (edge in 3 dimensions) when a relatively

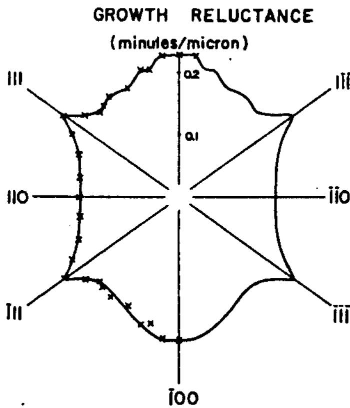


FIG. 4. Polar diagram of reciprocal normal dissolution rate for orientations of a ⟨110⟩ zone of germanium.

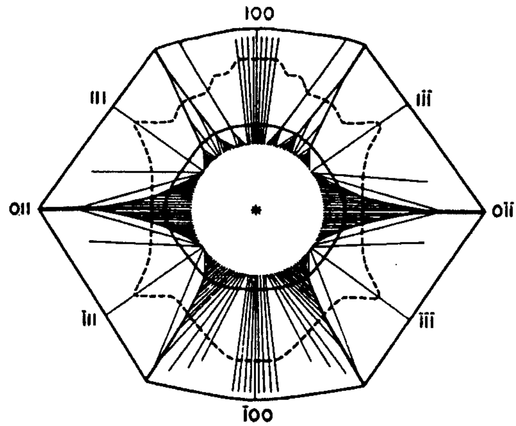


FIG. 5. Dissolution trajectories and derived profiles for the dissolution of a {110} circular section of germanium.

Figure 3.2: Frank's reluctance diagram and etch evolution [18].

minimum (or maximum) etch rate plane exists.

To determine the shape after etching (dissolution shape), Frank devised the reluctance diagram [17, 18]; he considered dissolution (etching) to correspond to growth reluctance. The reluctance diagram is a plot of the reciprocal of the etch rate versus crystal orientation, Figures 3.2, 3.3. It captures the properties of the etchant, process parameters, and substrate material. Frank demonstrated that the evolution of the shape of a smooth curve (of germanium under anisotropic etching) followed a path normal to the reluctance diagram at that orientation, and that orientation discontinuities (i.e., crystal edges) were formed when the normals to the reluctance diagram intersected. The edge formed by the etch process was parallel to the chord of the reluctance diagram between the intersecting paths. This method confirmed the findings of Batterman and was shown to agree with experiment.

The low aspect ratio etchants and smooth curvilinear masks used by Frank resulted in slowly evolving shapes, in which locally minimum (or maximum) planes eventually became dominant. Common etchants used on silicon wafers today (KOH, EDP, TMAH) exhibit far more anisotropy than the germanium experiments of Frank, Figures 3.1, 3.2, 3.3. Sequin extended Frank's approach and called the reciprocal plots *slowness diagrams* [51]. Sequin recognized that with high aspect ratio

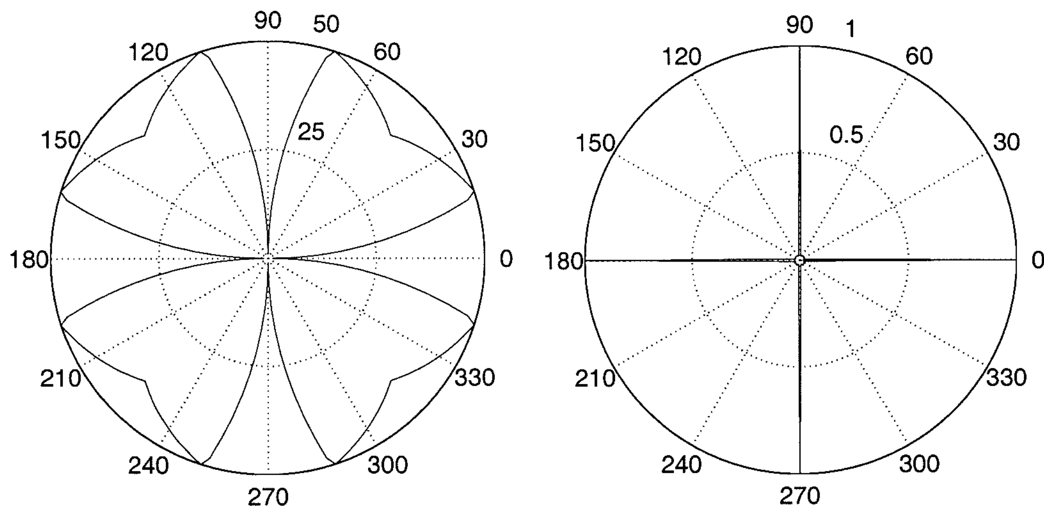


Figure 3.3: Etch rate and etch reluctance (slowness) diagram for KOH on silicon.

anisotropic processes, the local minimum or maximum planes would dominate immediately. The reluctance diagram could be used to examine all intermediate planes at a feature vertex and indicate which plane(s) would dominate (appear). Also by plotting the slowness (reluctance) vectors of the planes that formed a vertex, the trajectory of the vertex during the etch could be determined.

Sequin used a "circle of Thales" to compute the vertex velocity vector. This shows that the vertex velocity has a corresponding slowness vector that is normal to the chord between the slowness vectors of the planes forming the vertex [51] An alternative derivation is given below. Frank's reluctance diagram chord construction was used to derive a test for determining which (if any) planes would appear at a given vertex. Two-dimensional concave and convex vertices, formed by planes A and B, are shown in Figure 3.4 along with arbitrary reluctance (or slowness) diagrams. The vectors $S(A)$ and $S(B)$ are the slowness vectors corresponding to crystal edges A and B respectively. The portion of the reluctance diagram bounded by the slowness vectors of plane A and B is placed at the vertex.

For a concave vertex, the convex hull over the maxima of the reluctance plot (etch rate minima) and the chord between S(A) and S(B) is constructed. New vertices in the hull (S(C) and S(D)) correspond to the slowness vectors of planes

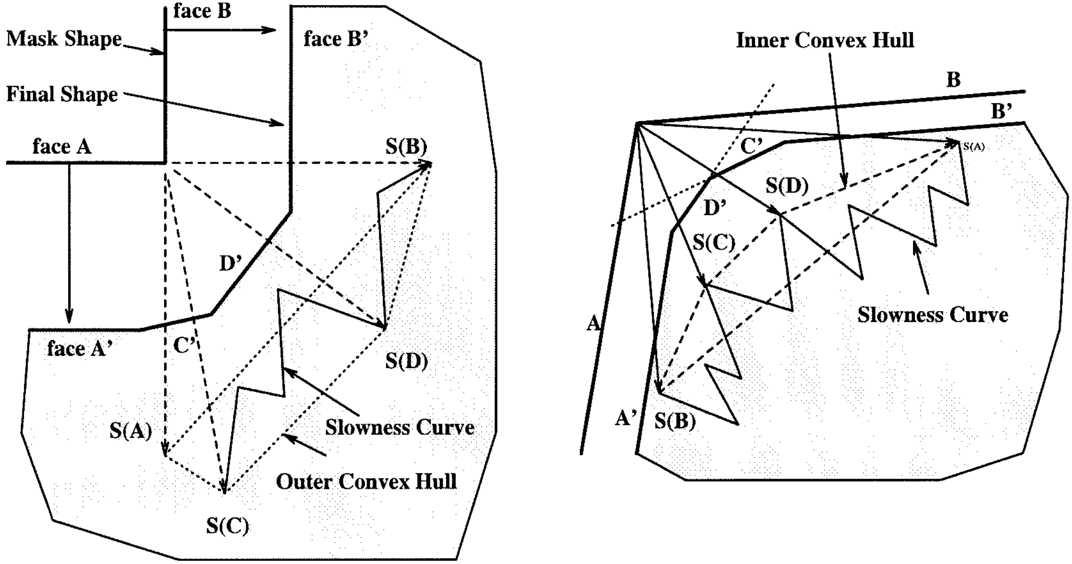


Figure 3.4: Convex hull construction over slowness (reluctance) diagram, after [51].

that will appear at the concave vertex. In this work, this is referred to as the *outer convex hull*, Figure 3.4.

Definition 3.1 Reluctance Vector*:* $R_i = \frac{1}{r_i} \cdot \hat{n}_i$

where r_i is the etch rate of the crystal plane with normal vector $\hat{n}_i$.

Definition 3.2 Reluctance Diagram *(set):* $D = \{R_i\}$

Definition 3.3 Outer Set*:* $\mathcal{O} = \{R_i \mid sgn[(\hat{n}_1 - R_i) \times (\hat{n}_1 - \hat{n}_2)] < 0,\ R_i \cdot \hat{n}_j > 0,\ j = 1, 2.\}$

Definition 3.4 Outer Convex Hull*:* $co\{\mathcal{O}, S_1, S_2\}$.

where, S_1 and S_2 are the reluctance (slowness) vectors of the two planes whose intersection form the vertex under consideration. See also Figure 3.5.

If the etch reluctance construction given above contributes new vertices to the outer convex hull, then new planes appear during the forward etch of the convex vertex. That is, if

$$| \ \partial(\text{co}\{\mathcal{O}, S_1, S_2\}) \ \cap \ \mathcal{O} \ | \ > 2$$

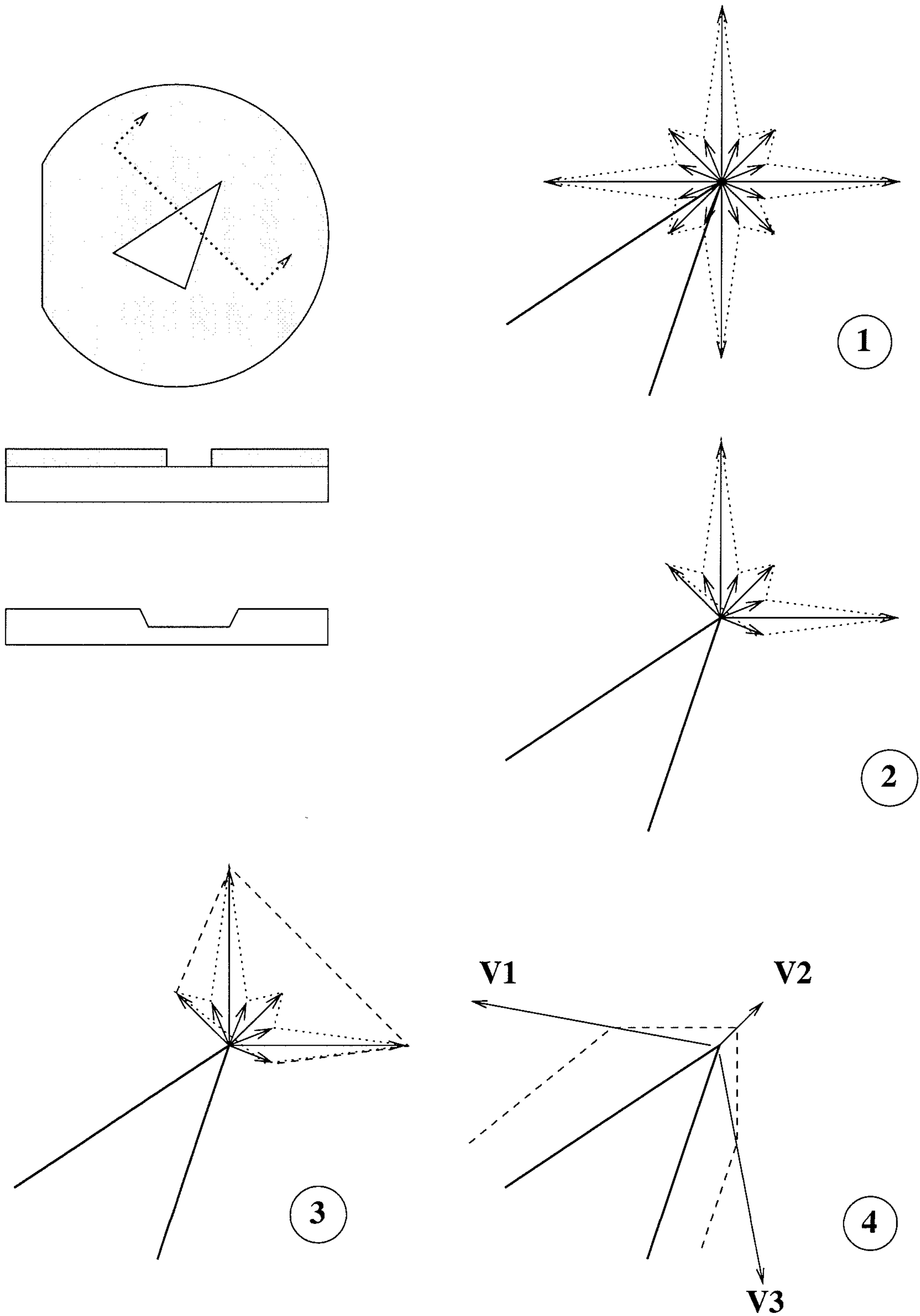


Figure 3.5: Convex hull construction over slowness (reluctance) diagram.

new planes will appear at this vertex during the etch.

For a convex vertex, the convex hull over the minima of the reluctance plot (etch rate maxima) and the chord between S(A) and S(B) is constructed. New vertices in the hull (S(C) and S(D)) correspond to the slowness vectors of planes that will appear at the convex vertex [51]. Similarly, this is referred to as the *inner convex hull*, Figure 3.4.

Definition 3.5 Inner Set: $\mathcal{I} = \{R_i \mid sgn[(\hat{n}_1 - R_i) \times (\hat{n}_1 - \hat{n}_2)] > 0,\ R_i \cdot \hat{n}_j > 0,\ j = 1, 2.\}$

Definition 3.6 Inner Convex Hull: $co\{\mathcal{I}, S_1, S_2\}$.

If the etch reluctance construction above contributes new vertices to the inner convex hull, then new planes appear during the forward etch of the concave vertex. That is, if

$$| \ \partial(\text{co}\{\mathcal{I}, S(A), S(B)\}) \ \cap \ \mathcal{I} \ | \ > 2$$

new planes will appear at this vertex during the etch.

For etchants/materials with low aspect ratio anisotropy or pseudo-isotropic behavior, the convex hulls include smooth regions of the reluctance diagram. Thus a span of planes is introduced resulting in a smooth curve in the etch front. Sequin [51] has also extended these methods to three dimensions.

With these developments, the above model yields an etch rate for a given edge, a trajectory for the vertex formed by adjacent edges, and a test to determine which if any planes will appear during the etching process.

3.1.1 Geometric Etch Computation

Given planes i and $i+1$, which form vertex i (Figure 3.6), the rate of etch of the crystal planes, r_i and r_{i+1}, are given by the etch rate diagram. The rate at which

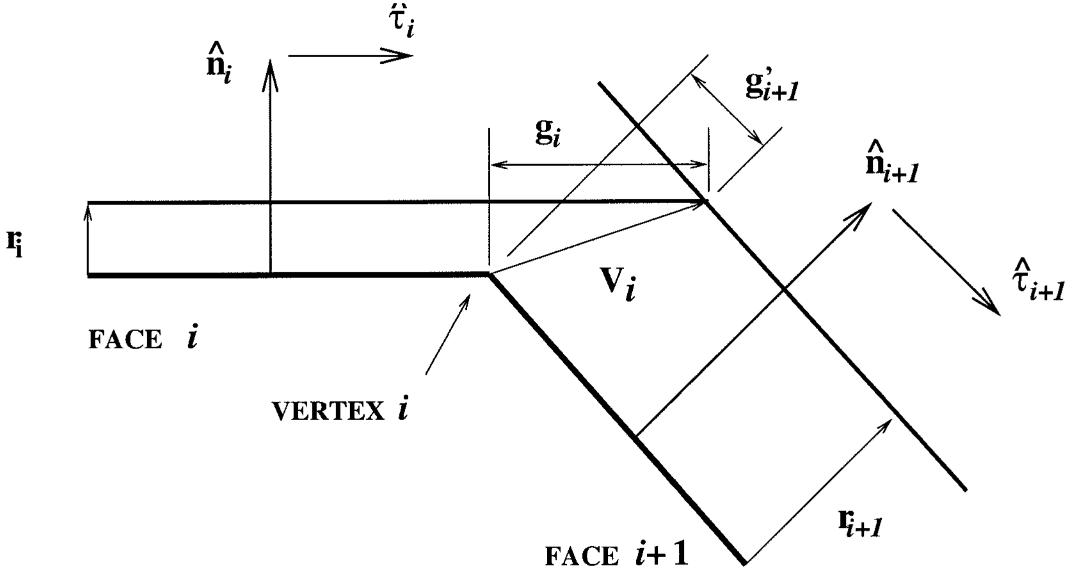


Figure 3.6: Vertex velocity computations diagram.

plane i grows (increases length), relative to the normal to edge i at vertex i is given by g_i. Thus, the vertex velocity, V_i, can be written:

$$
\begin{aligned}
V_i &= r_i \cdot \hat{n}_i + g_i \cdot \hat{\tau}_i && (3.1) \\
&= r_{i+1} \cdot \hat{n}_{i+1} + g'_{i+1} \cdot \hat{\tau}_{i+1} && (3.2)
\end{aligned}
$$

where $\hat{n}_i$ and $\hat{\tau}_i$ are the normal and tangential vectors to plane i, and r_i is the etch rate of plane i. V_i represents the rate and direction of vertex advancement during the etch. Note that g_i represents the growth of plane i due to vertex i only. The growth of plane $i+1$ due to vertex i is given by $-g'_{i+1}$. Equating the plane i and $i+1$ vertex velocities and multiplying by $\hat{n}_{i+1}$ yields:

$$
\begin{aligned}
g_i \cdot \hat{\tau}_i - g'_{i+1} \cdot \hat{\tau}_{i+1} &= r_{1+i} \cdot \hat{n}_{i+1} - r_i \cdot \hat{n}_i \\
g_i(\hat{\tau}_i \cdot \hat{n}_{i+1}) &= r_{i+1} - r_i(\hat{n}_i \cdot \hat{n}_{i+1})
\end{aligned}
$$

$$g_i = \frac{r_{i+1} - r_i(\hat{n}_i \cdot \hat{n}_{i+1})}{\hat{\tau}_i \cdot \hat{n}_{i+1}} \tag{3.3}$$

similarly for g'_{i+1}:

$$g'_{i+1} = \frac{r_i - r_{i+1}(\hat{n}_{i+1} \cdot \hat{n}_i)}{\hat{\tau}_{i+1} \cdot \hat{n}_i} \tag{3.4}$$

Thus,

$$\begin{aligned} V_i &= r_i \cdot \hat{n}_i + [\frac{r_{i+1} - r_i \hat{n}_i \cdot \hat{n}_{i+1}}{\hat{\tau}_i \cdot \hat{n}_{i+1}}] \cdot \hat{\tau}_i \\ &= r_{i+1} \cdot \hat{n}_{i+1} + [\frac{r_i - r_{i+1}\hat{n}_{i+1} \cdot \hat{n}_i}{\hat{\tau}_{i+1} \cdot \hat{n}_i}] \cdot \hat{\tau}_{i+1} \end{aligned} \tag{3.5}$$

where $\hat{n}_i$ and $\hat{\tau}_i$ are the normal and tangential vectors to plane i, and r_i is the etch rate of plane i.

As stated above a new plane may appear at certain vertices. In effect, the vertex will "split" into two or more vertices. Equation 3.5 is extended for the case of one or more appearing planes. The single appearing plane vertex velocities are given by:

$$V_{i,1} = r_i \hat{n}_i + \frac{r'_1 - r_i(\hat{n}_i \cdot \hat{n}'_1)}{\hat{n}'_1 \cdot \hat{\tau}_i} \cdot \hat{\tau}_i \tag{3.6}$$

$$V_{i,2} = r'_1 \hat{n}'_1 + \frac{r_{i+1} - r'_1(\hat{n}'_1 \cdot \hat{n}_{i+1})}{\hat{n}_{i+1} \cdot \hat{\tau}'_1} \cdot \hat{\tau}_1 \tag{3.7}$$

where r'_1, $\hat{n}'_1$, and $\hat{\tau}'_1$ are the etch rate, normal, and tangent unit vectors of the appearing plane at vertex i.

Often it is useful to know when two vertices might merge and a plane will disappear, or the rate at which a plane is growing. Given edge i bounded by vertices at positions P_i and P_{i+1}, with normal $\hat{n}_i$, tangent $\hat{\tau}_i$, and length l_i, the sum of the growth rates at both vertices defining the plane give the rate of growth of plane i :

$$
\begin{aligned}
G_i &= g_i - g_i' \\
&= \frac{r_{i+1} - r_i(\hat{n}_i \cdot \hat{n}_{i+1})}{\hat{\tau}_i \cdot \hat{n}_{i+1}} - \frac{r_{i-1} - r_i(\hat{n}_i \cdot \hat{n}_{i-1})}{\hat{\tau}_i \cdot \hat{n}_{i-1}} \qquad (3.8)
\end{aligned}
$$

then,

$$
\begin{aligned}
P_i + V_i \cdot T^* &= P_{i-1} + V_{i-1} \cdot T^* \\
P_i - P_{i-1} &= l_i \cdot \hat{\tau}_i \\
l_i \cdot \hat{\tau}_i &= (V_{i-1} - V_i) \cdot \Delta t \qquad (3.9)
\end{aligned}
$$

where T^* is the etch time at which the vertices i and $i+1$ meet and the length of plane i goes to zero, e.g., the plane vanishes.

If $G_i < 0$ use the two forms of Equation 3.1 to compute the time, T_i^*, at which plane i will disappear:

$$
\begin{aligned}
l_i \cdot \hat{\tau}_i &= (V_{i-1} - V_i) \cdot T^* \\
&= (r_i \cdot \hat{n}_i + g_i' \cdot \hat{\tau}_i - r_i \cdot \hat{n}_i - g_i \cdot \hat{\tau}_i) \cdot T^* \\
l_i &= (g_i' - g_i) \cdot \Delta t \qquad (3.10)
\end{aligned}
$$

thus,

$$
\begin{aligned}
T_i^* &= l_i \cdot (g_i' - g_i)^{-1} \\
T_i^* &= l_i \cdot (-G_i)^{-1} \text{ when } (G_i < 0) \qquad (3.11)
\end{aligned}
$$

The calculations of vertex velocities, appearing planes (edges), growth rates, and disappearance times, provide the tools required to simulate the material removal process. By calculating the vertex velocity, the evolution of a vertex can be computed by

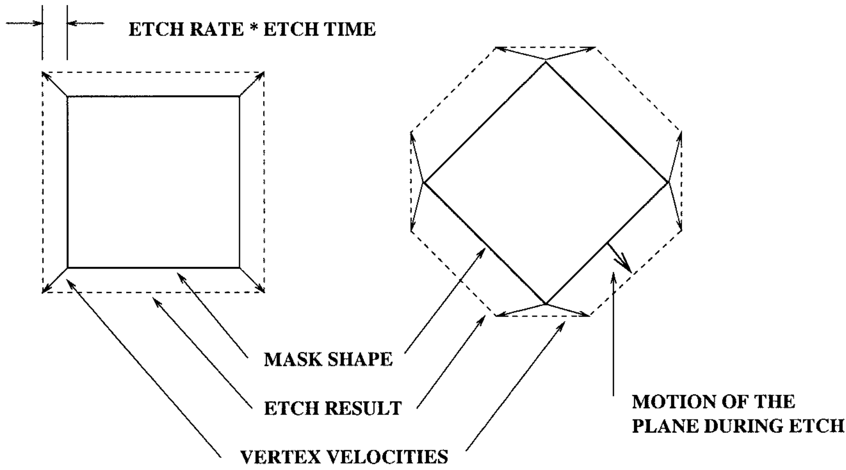


Figure 3.7: Vertex velocity example.

$$P_i(t) = V_i\ t \tag{3.12}$$

where $P_i(t)$ is the vector of position coordinates of vertex i at time t, V_i is the velocity of vertex i computed in Equation 3.5, and t is the time of the etch. This is referred to as *forward* simulation (or *forward* etch), since the etch is proceeding in forward time. In Figure 3.7 the evolution of vertices during forward simulation is shown; these shapes are taken from Figures 2.3 and 2.4.

3.2 Mask Layout Approach

The proceeding section provided the means to calculate vertex velocities and the evolution of shapes under wet etching. One might ask why this simulation cannot be run in *reverse time* to arrive at the desired mask layout geometry for a desired output shape. i.e., would $t < 0$ in Equation 3.12 correspond to a vertex location in the mask layout for a desired shape?

The desired shapes are composed of planes and vertices. For a plane, its normal velocity is described by the etch rate diagram; therefore, given the location of a plane

in the desired output, the location of the plane in the mask layout is known to be at a distance given by the plane's etch rate and the planned etch time, Figure 3.7. However, one must account for motion of the bounding vertices that define the plane. Vertices cannot always be trivially reverse simulated.

If no new planes appear at the intersection of two edges, then this simple "reverse simulation" may be used.

$$P_i(-t) = P_i(0) - V_i\ t \tag{3.13}$$

where the etch evolves from $0 \rightarrow -T$ from the desired geometry ($t = 0$) to the mask layout ($t = -T$).

However, it is possible for an infinite number of different planes to have existed in the mask layout that will disappear before the desired vertex is etched. Thus, etching is an infinite to one mapping, and having a multiplicity of solutions to any vertex will have value.

This simple approach of *reverse simulation* results in a local mask shape that is topologically identical to the desired output, Figure 3.8. If the vertex has new crystal planes that appear at $t = 0^+$, then any topologically similar mask layout, regardless of location or scale, will result in a shape with these undesired planes, Figure 3.9. (See also Figure 2.4). Therefore, a mask layout that can resist or delay the appearance of these new planes is required. In order to do so, a more sophisticated method must be developed.

It is evident that vertices with appearing planes need a different class of solution. To begin, a few definitions are in order.

Definition 3.7 Simple Vertex*: A vertex in which no crystal planes appear during the forward etch.*

$$\text{Simple Concave Vertex: } |\ \partial(\text{co}\{\mathcal{O}, S_1, S_2\}) \cap \mathcal{O}\ | = 2 \tag{3.14}$$

$$\text{Simple Convex Vertex: } |\ \partial(\text{co}\{\mathcal{I}, S_1, S_2\}) \cap \mathcal{I}\ | = 2 \tag{3.15}$$

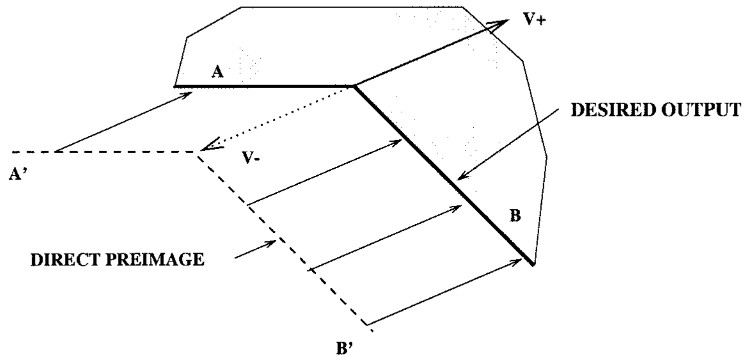


Figure 3.8: "Reverse simulation" of etching.

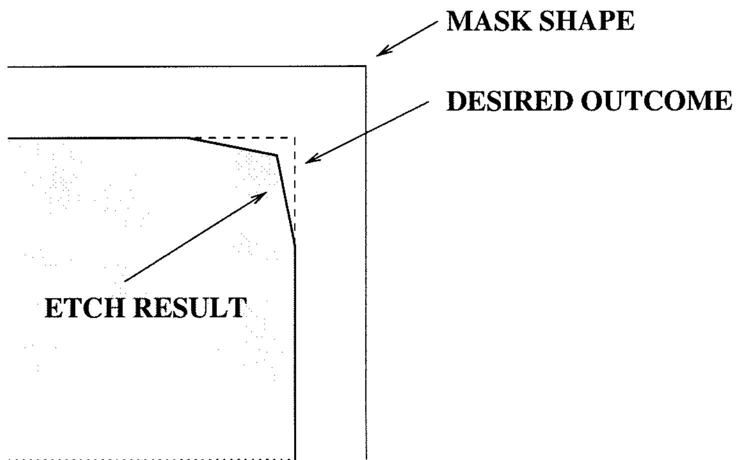


Figure 3.9: Appearing planes during etch.

Definition 3.8 Compound Vertex*: A vertex in which one or more crystal planes appear during the forward etch.*

$$\text{Compound Concave Vertex: } |\ \partial(\text{co}\{\mathcal{O}, S_1, S_2\}) \cap \mathcal{O}\ | \ > \ 2 \qquad (3.16)$$

$$\text{Compound Convex Vertex: } |\ \partial(\text{co}\{\mathcal{I}, S_1, S_2\}) \cap \mathcal{I}\ | \ > \ 2 \qquad (3.17)$$

Definition 3.9 Vertex Preimage*: The portion of the mask layout that etches to a specific vertex shape.*

Definition 3.10 Direct Preimage*: A vertex preimage computed using a reverse simulation of the vertex forward etch velocity.*

Definition 3.11 Indirect Preimage*: Any non-direct vertex preimage.*

Vertices can now be classified according to this taxonomy and solutions created for each.

The general idea underlying the construction of an indirect preimage is to infinitesimally modify the local topology at a vertex such that a single vertex is replaced with multiple *simple* vertices. A direct preimage is then computed for each new vertex, Figure 3.10. That is, some number of new *seed* planes, each of infinitesimal size, are inserted into the shape and a simulation is run backwards in time to obtain a mask shape. When the mask is subsequently etched, the inserted planes will disappear exactly as the etch is concluded, Figure 3.11. Figure 3.12 summarizes the structure of the approach for simple concave vertices.

While the indirect preimage modifies the topology of the resultant vertex, it is also possible to stop the reverse simulation at points along the way and make further modifications, via additional plane insertions. When to pause the reverse simulation and which planes to insert, rely on timing models presented in Chapter 5, Figure 3.11.

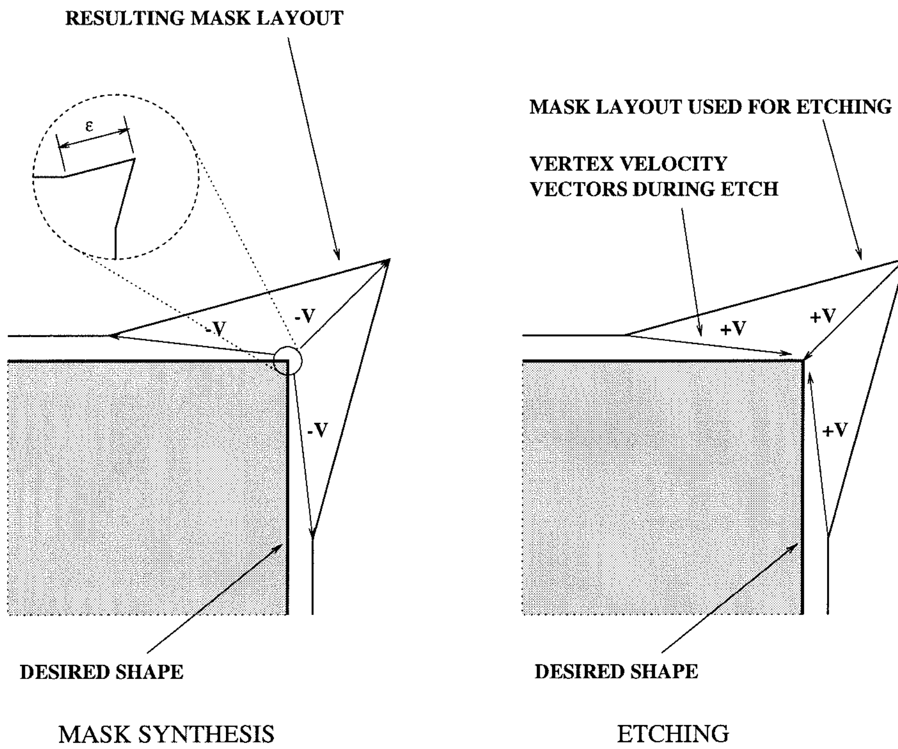


Figure 3.10: Example of topology modification and reverse simulation.

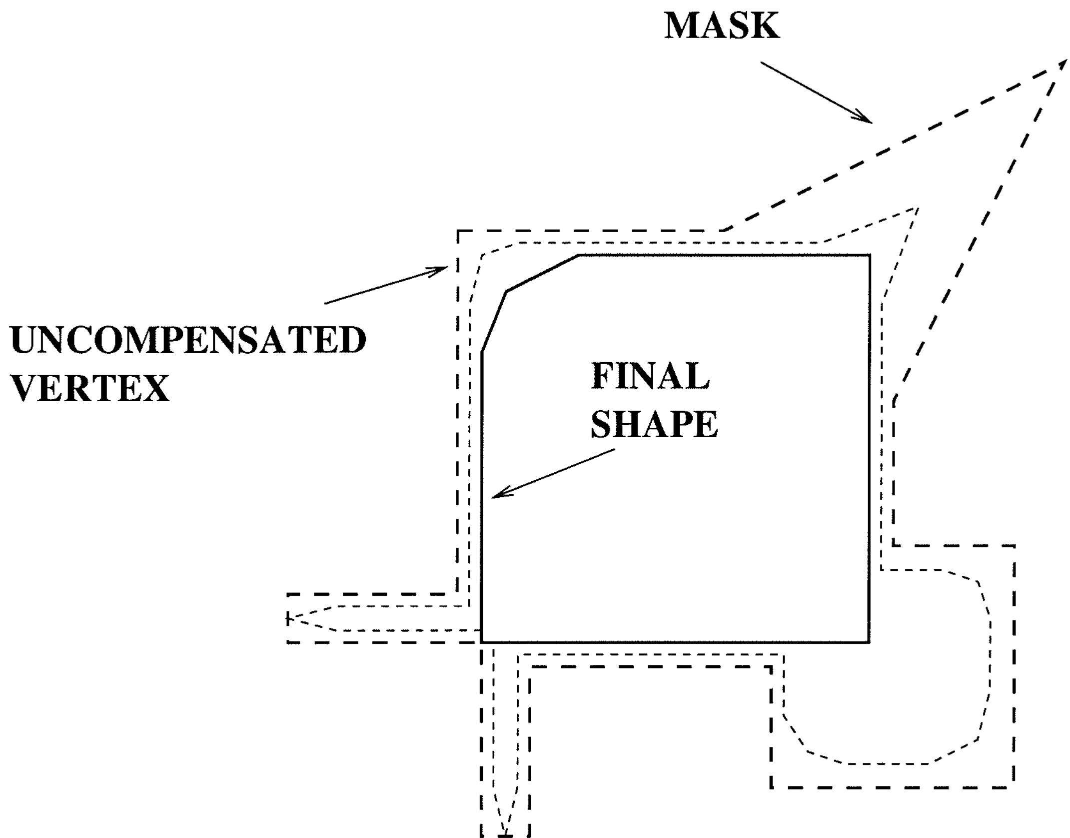


Figure 3.11: Example of preimages for compound convex vertices, i.e., compensation structures.

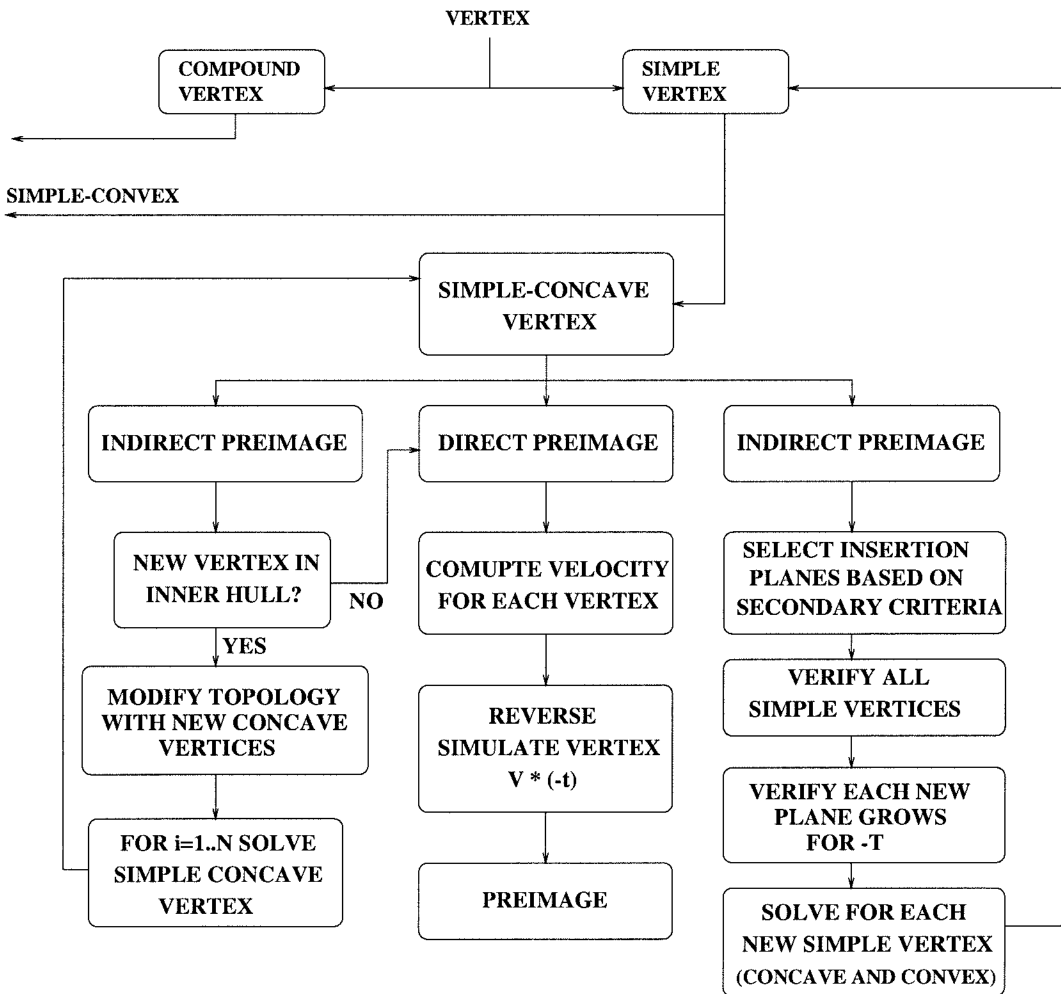


Figure 3.12: Structure of simple concave solution approach.

3.3 Summary

To summarize the approach, a set of techniques has been developed to analyze each vertex in the desired output of a wet etching process and create a preimage for each vertex type. In general, the approach relies upon the modification of the desired shape topology through the insertion of infinitesimal planes, such that each new feature vertex is *simple* and its preimage may be computed using direct reverse simulation. The methods describe a family of solutions for both simple and compound vertices and the topology modification can occur at any point along the duration of the reverse simulation.

The next chapter presents methods to compute a preimage for each of the four types of vertices, *simple-concave*, *simple-convex*, *compound-concave*, and *compound-convex.*

Chapter 4

Mask Synthesis

This chapter describes the development of methods to compute the mask layout geometry for a given bulk etch micro-machining process. That is, given a desired part geometry and process parameters, the synthesis procedure determines a candidate mask geometry that will etch to the final desired shape, even in the case of highly anisotropic etchants. While these methods are not inherently restricted to planar polygons, the algorithm development is presented for two-dimensional polygonal geometry in order to provide a clear exposition. Extensions to three dimensions are briefly discussed in Chapter 8.

The polygonal geometry of the desired parts is composed of edges and vertices. An edge or plane section is etched at a rate described by the etch rate diagram (ERD). However, due to the anisotropy of the etchant, vertex evolution may exhibit quite complex behavior during the etch process, with some vertices being eliminated and others appearing. Since the behavior of planes during the etch can be determined from the ERD, the challenge for mask synthesis is to determine the preimage of each vertex.

The mask layout synthesis process is based on the decomposition of the desired part shape into primitive features, which are 2-D vertices in the case of planar polygonal objects. Vertices can be categorized as either concave or convex in the usual way. The vertices where new planes appear during the forward-time etching process are categorized as *compound* and those vertices in which no planes appear

as *simple.* The method of this classification was described in § 3.2.

Note that the compound/simple categorization is a function of the local geometry, its relative orientation to the crystal, and the etchant/process properties. This taxonomy will be very useful in the computation of mask shapes for desired output geometry. The mask geometry that produces a desired feature of the output is called its *preimage.* Further, a *direct preimage* is derived from the application of the reverse simulation to a vertex. Since there are multiple preimages for a desired output, an alternative *indirect preimage* may be derived by locally altering the vertex topology to contain multiple vertices, and then computing direct preimages for each of the new vertices.

While the general synthesis approach is based on the concept of reverse simulation, in some cases the local topology of the evolving mask is altered before or during the reverse simulation.

A method to compute the preimage for each of the four types of local geometric features, *simple-concave, simple-convex, compound-concave*, and *compound-convex*, is now presented. The "feature preimage" is a local mask shape that when etched for the specified time will produce the desired single vertex in the specified position and orientation.

4.1 Simple Concave Vertices

Simple-concave vertices have multiple preimages which will etch to the desired shape. Therefore, techniques for computing multiple preimages are developed. Secondary considerations may then be used to select a given solution from the range of possible preimages.

4.1.1 Simple Concave Direct Preimage

For the *simple-concave* vertex there exists a *direct preimage.* The direct preimage is formed by running the forward simulation in reverse-time without consideration of available topology alterations. In Section 3.1.1, Equation 3.5 was shown to describe

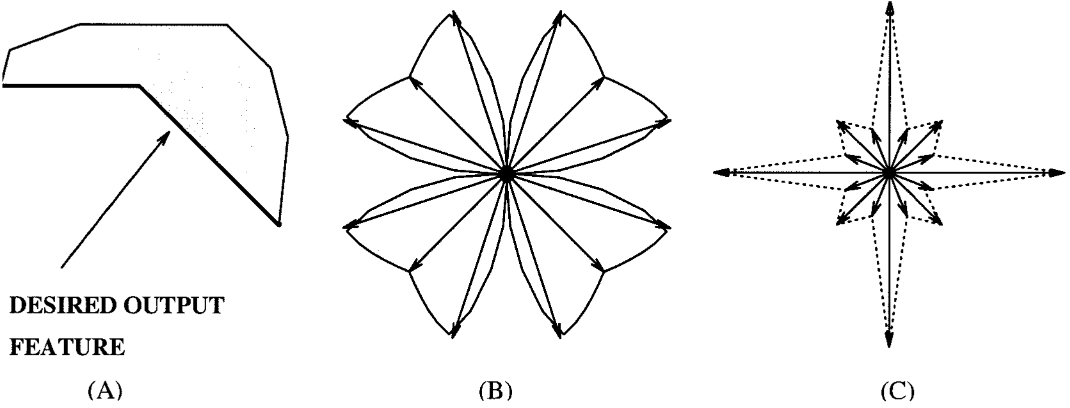


Figure 4.1: (A) Concave vertex, (B) Etch rate diagram, (C) Reluctance (Slowness) diagram.

the trajectory of a simple concave vertex. The direct preimage is locally topologically equivalent to the target vertex. Thus the position of the preimage vertex can be calculated using the simple formula:

$$P_{preimage} = P_{desired} - (V * t) \tag{4.1}$$

where $P_{preimage}$ is the position of the vertex in the mask layout, $P_{desired}$ is the desired location of the vertex in the final part geometry, t is the etch time, and V is the vertex velocity vector.

The procedure for *simple-concave* vertices is demonstrated by example. A concave corner, the etch rate, and reluctance diagrams for a KOH-like process is shown in Figure 4.1. Figure 4.2 shows the reluctance vectors, $S(A)$ and $S(B)$, for the two faces forming the vertex, the included portion of the reluctance diagram, and the outer convex hull of the reluctance vectors. This diagram verifies that the reluctance diagram does not form any vertices in the outer convex hull and therefore this vertex is *simple*.

The vertex velocity vector is computed according to Equation 3.1 and the location of the preimage vertex is given by $P + V * (-t)$, where P is the position of the desired vertex.

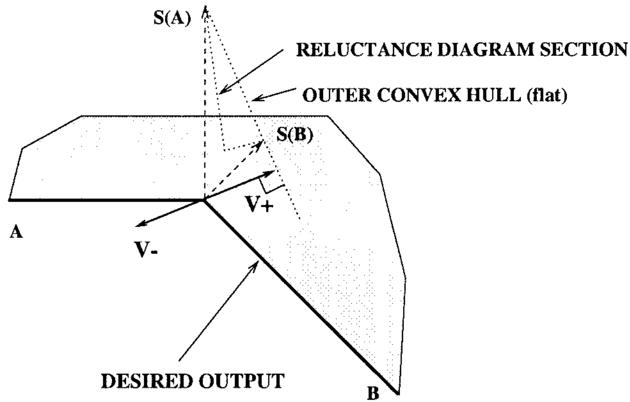


Figure 4.2: Convex hull construction over reluctance diagram.

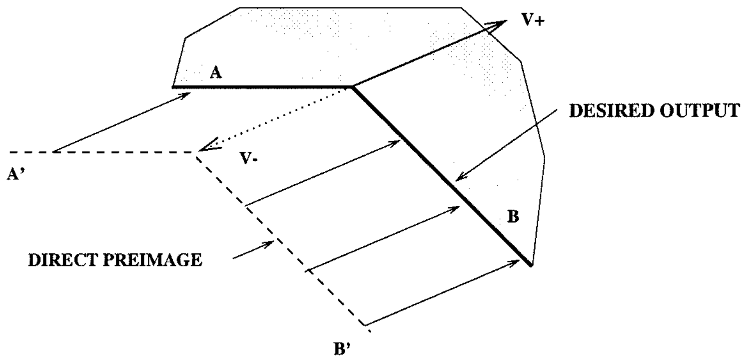


Figure 4.3: Direct preimage construction.

Figure 4.3 shows preimage planes A' and B' which etch forward to the required shape given by planes A and B.

While the direct preimage provides a single solution, there exists other valid preimages which etch the desired vertex.

4.1.2 Simple-Concave Inner-Hull Indirect Preimage

The concept behind the method to create an indirect preimage for a simple concave vertex is to find additional planes that may exist in the preimage that will disappear at or before the conclusion of the etch at time $t = \tau$. A plane disappears when the two neighboring vertices join into one (there two forward vertex velocity vectors

intersect). The vertices of the local preimage for this vertex will have vertex velocity vectors that all meet at $t = \tau$ at the desired vertex location. The case where the vertex velocity vectors meet at some time other than $t = \tau$ will be treated in the next chapter.

In most cases there are an infinite number of solutions that will meet this criteria. To compute an outer bound for this class of solutions, the planes whose orientations lie within the included angle of the vertex and have the locally highest etch rate are selected, Figure 4.4. These planes are readily determined by computing the inner-hull of the local reluctance diagram. The planes which contribute new vertices to the convex hull represent the planes with the locally highest etch rate.

This preimage solution is depicted in two different ways to give the reader geometrical insight into the solution. Figure 4.4 shows the derivation of multiple inverse velocity vectors from the reluctance diagram. Note that the forward vertex velocity vectors are computed using the outer-hull of the reluctance diagram, whereas the inverse solution is born from the inner-hull. The alternative conceptualization of the solution is seen in Figure 4.5. Here the inner hull is used to determine how one can feasibly alter the local topology of the feature vertex by inserting new planes corresponding to the vertices of the inner-hull. Each new plane is infinitesimal in size and admits one new vertex to the feature topology. A direct preimage for each new vertex in the compound feature is then computed using the methods of § 4.1.1.

In some cases the inner-hull of the local reluctance diagram does not contribute a vertex to the convex hull, then this method does not contribute additional preimage solutions.

For the current example, Equation 3.5 is used to compute the multiple vertex velocity vectors, denoted by $\mathbf{V}_{i,j}$ in Figure 4.4, where i is the vertex number and $j \in 1, k$ refers to one of the k inverse vertex velocity vectors. The two inverse vertex velocity vectors computed by:

$$\mathbf{V}_{i,1} = r_i \hat{\mathbf{n}}_i + \frac{r'_1 - r_i(\hat{\mathbf{n}}_i \cdot \hat{\mathbf{n}}'_1)}{\hat{\mathbf{n}}'_1 \cdot \hat{\tau}_i} \cdot \hat{\tau}_i \tag{4.2}$$

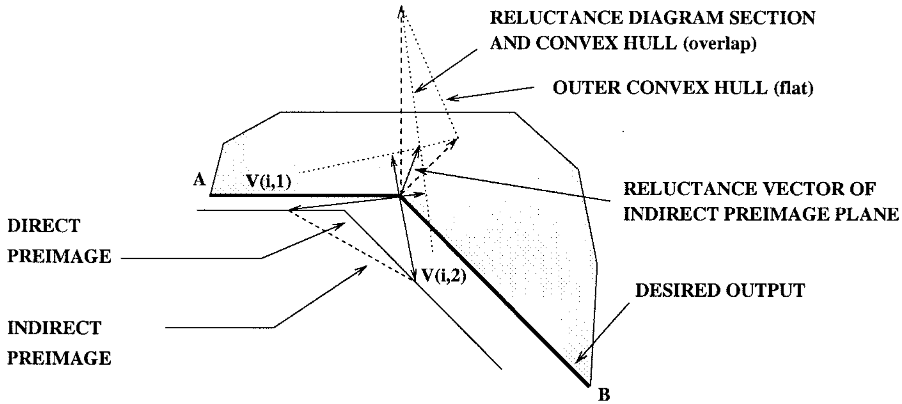


Figure 4.4: Indirect preimage construction with comparison to direct preimage.

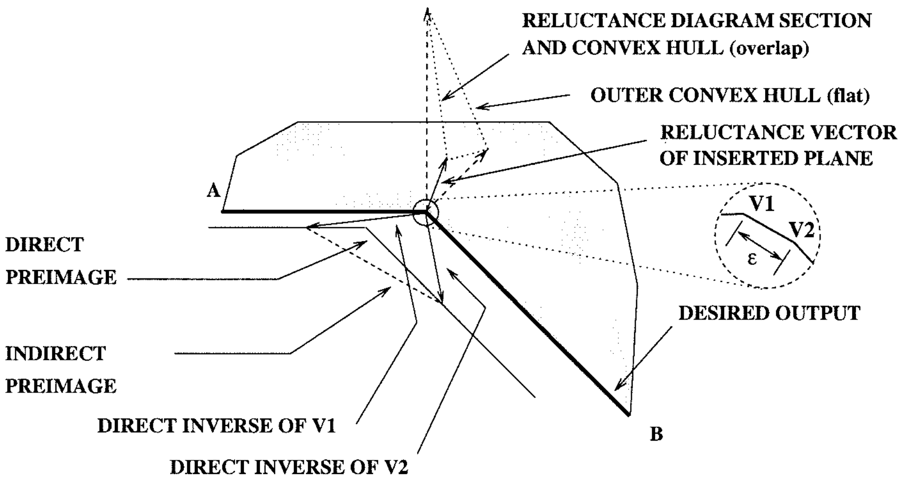


Figure 4.5: Topology modification during indirect preimage construction.

$$\mathbf{V}_{i,2} = r'_1\hat{\mathbf{n}}'_1 + \frac{r_{i+1} - r'_1(\hat{\mathbf{n}}'_1 \cdot \hat{\mathbf{n}}_{i+1})}{\hat{\mathbf{n}}_{i+1} \cdot \hat{\tau}'_1} \cdot \hat{\tau}`_1 \quad (4.3)$$

where r_j is the etch rate associated with edge e_j, and $\hat{\tau}_j$ and $\hat{\mathbf{n}}_j$ respectively denote the unit vectors tangent and normal to e_j. r'_1,$\hat{\tau}'_1$, and $\hat{\mathbf{n}}'_1$ are the respective values for the new appearing plane (vertices of the inner convex hull).

Notice that the method presented in this section examines only those planes whose normal lies in the interior of the cone spanned by the normals of neighboring planes as candidates for inclusion. Thus all new vertices are also concave. In the next section this restriction is lifted, and examples of other candidate preimages are developed.

4.1.3 Alternative Indirect Simple-Concave Preimages

The indirect preimage solutions in the previous section were limited to the insertion of those planes whose normal lies in the interior of cone spanned by the normals of the neighboring planes. In this section that restriction is lifted and other methods to create indirect preimages are examined.

A simple-concave vertex is shown in Figure 4.6 formed by plane A and B. In this case, selection from among the available planes for inclusion in the indirect preimage will be based upon the magnitude of the tangential component of the vertex velocity along plane A. That is, which plane, when inserted into the topology, and used to compute a preimage, will result in preimage vertices with minimum velocity tangent to plane A and result in simple vertices that may be solved with the methods described above. By minimizing the tangential component of the vertex velocity, the interaction of this vertex's preimage with neighboring vertices can be delayed or eliminated.

In order to determine which plane, when inserted into the topology, will minimize the component of the vertex velocity along plane A, the γ-*velocity graph* is generated, Figure 4.7. The vertical axis of the plot corresponds to the magnitude of component of the vertex velocity vector along the tangent to plane A. The horizontal axis corresponds to angle of the inserted plane, γ. Note that the vertex is measured

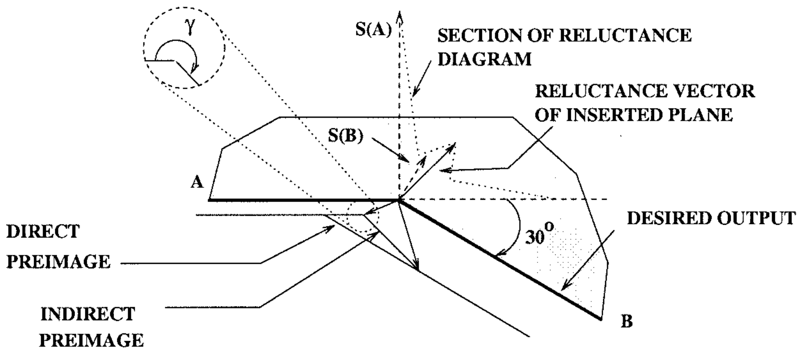


Figure 4.6: Alternative indirect preimage construction.

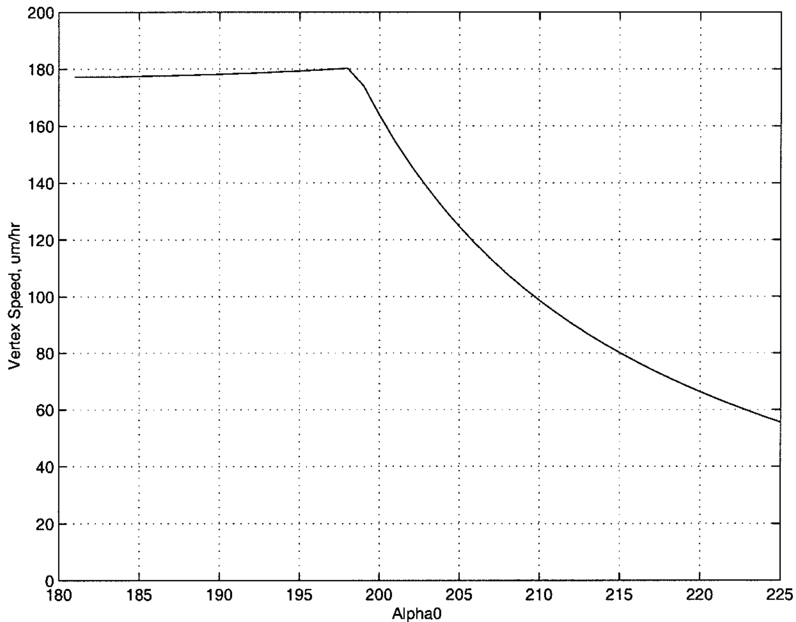


Figure 4.7: γ-Velocity Plot.

relative to plane A, Figure 4.6, thus $\gamma \in [180^o, 360^o]$.

From Figure 4.7, a plane of angle 225° results in the minimum tangential vertex velocity. This is a function of the orientations of the two planes and the process parameters. Values of $\gamma > 225$ result in compound vertices which require a different solution procedure (§ 4.4); so only $180^o < \gamma < 225^o$ is displayed. The desire to minimize the value of γ must be balanced against the robustness of the mask layout,

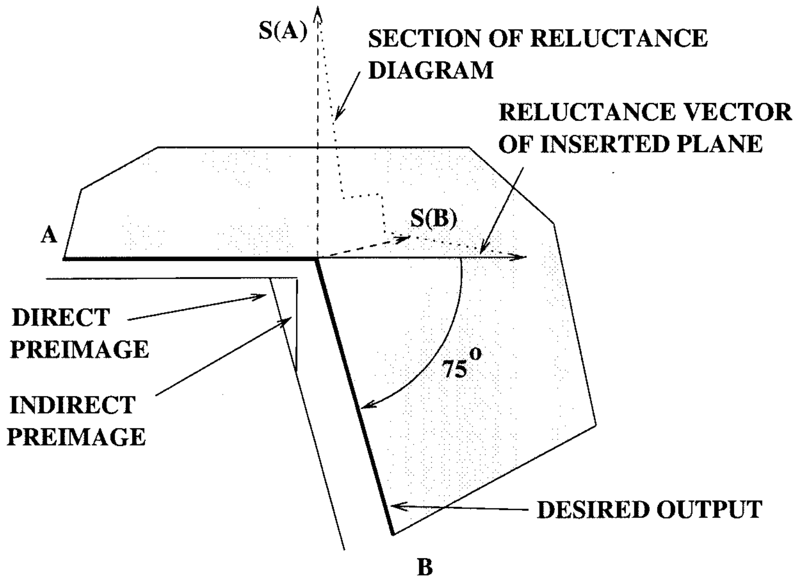


Figure 4.8: Alternative indirect preimage construction.

since slight misalignment could cause a simple vertex to become compound resulting in an unintended result shape. Figure 4.8 shows similar solutions for other relative plane angles.

4.2 Simple Convex

The process of computing the preimage for the *simple-convex* vertex is analogous to the *simple-concave* case discussed in § 4.1. Convex and concave vertices use opposite convex hulls in their analysis. Where the concave vertex solution relied upon the outer hull, the convex vertex solution will utilize the inner hull.

4.2.1 Simple Convex Direct Preimage

Figure 4.9 shows the convex vertex with the etch rate diagram and reluctance diagram. In Figure 4.10 the reluctance vectors for the two planes are shown with the included portion of the reluctance diagram and the convex hull. Notice that in the case of a convex vertex the "inner" convex hull is checked and it is determined that there are no vertices of the hull contributed by the reluctance diagram. Thus the

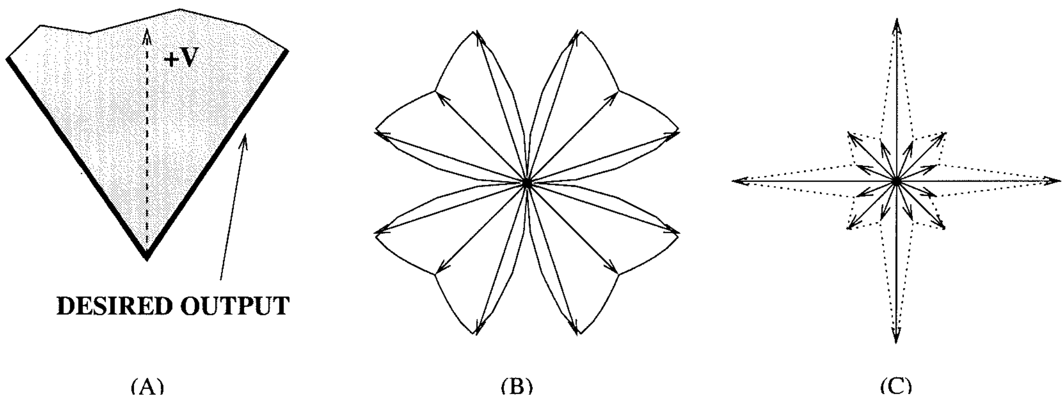


Figure 4.9: (A) Convex vertex, (B) Etch rate diagram, (C) Reluctance (Slowness) diagram.

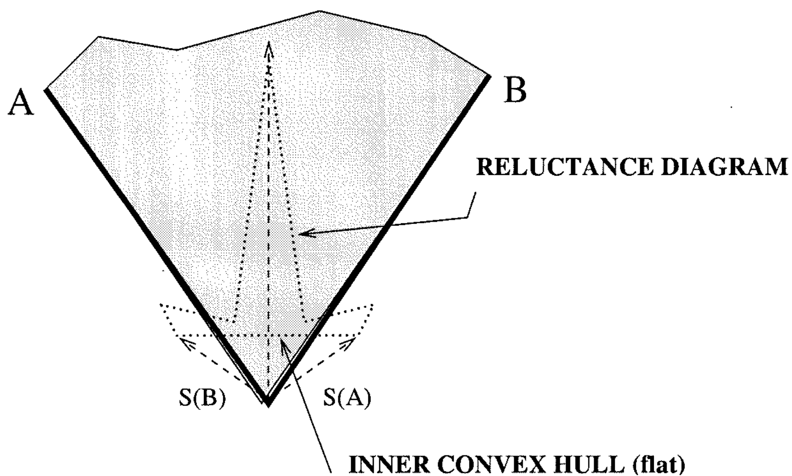


Figure 4.10: Convex vertex overlaid with reluctance diagram.

vertex is indeed simple.

Equation (3.5) is used to compute the vertex velocity vector. Its sign is reversed and the direct preimage is computed in the usual way. See Figure 4.11.

4.2.2 Simple Convex Outer-Hull Preimage

The computation of an outer-hull preimage is similar to the inner-hull preimage procedure for the *simple-concave* vertex discussed in § 4.1.2. The "outer" convex

Figure 4.11: Direct preimage construction.

Figure 4.12: Indirect preimage construction.

hull is used to determine which planes have the minimum velocity normal to the vertex. These planes are used to modify the local topology and to compute the multiple inverse vertex velocity vectors $\mathbf{V}_{i,j}$ using Equation 4.2 as shown in Figure 4.12. While the indirect preimage of § 4.1.2 resulted in a fast plane inserted into the topology, the convex indirect solution results in the slowest plane possible being inserted. The choice of the slowest plane results in a preimage with minimum distance between the mask vertex and the corresponding desired feature vertex, thus saving space on the wafer.

4.3 Compound Convex

Unfortunately, the simple techniques that were employed for the *simple* vertices cannot be applied to compound ones. A *compound* vertex, by definition, splits into two or more vertices during etch progression, and thus has multiple forward vertex velocity vectors and appearing plane(s). However, the desired resultant topology does not contain this appearing plane. Since any direct preimage for this vertex (being topologically identical) will create this plane at time $t = 0^+$, the proper vertex will not be etched, Figure 3.11.

Figure 4.13 shows a convex vertex with the associated etch rate and reluctance diagrams. The reluctance vectors of the two faces are shown in Figure 4.14 with the included portion of the reluctance diagram and the "inner" convex hull. The vertices in the hull due to the reluctance curve correspond to the new planes that will appear in the forward etch of this shape. Thus, this is indeed a *compound* vertex.

This situation can also be examined by looking at the vertex velocity. If the simple direct preimage method were used, the vectors would cross each other at $t = 0^-$ and the resulting preimage geometry would be physically unrealizable (Figure 4.13(A)). Therefore, in the same way that an indirect preimage for simple vertices was created, a physically realizable indirect preimage that will form the desired *compound-concave* vertex must be devised.

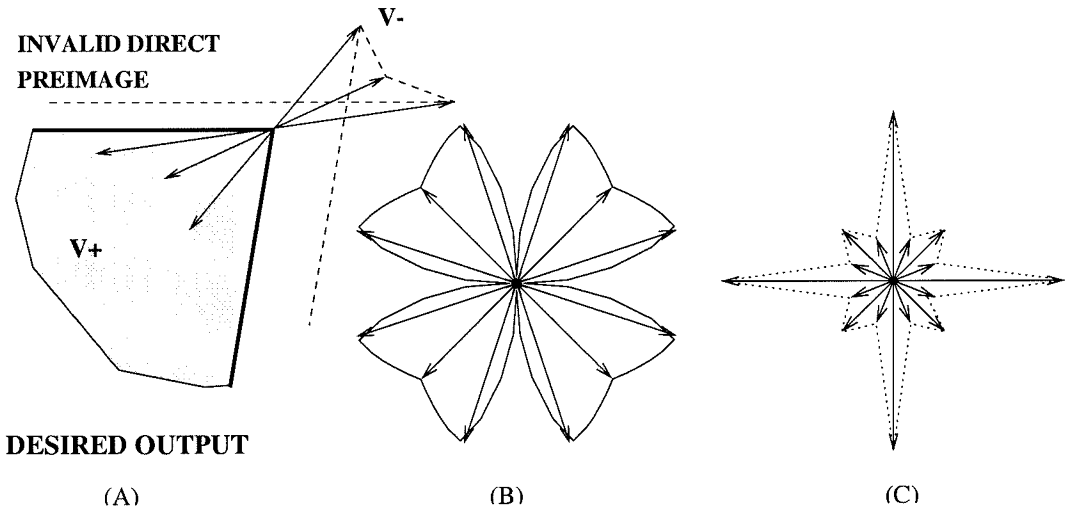


Figure 4.13: (A) Compound convex vertex, (B) Etch rate diagram, (C) Reluctance diagram.

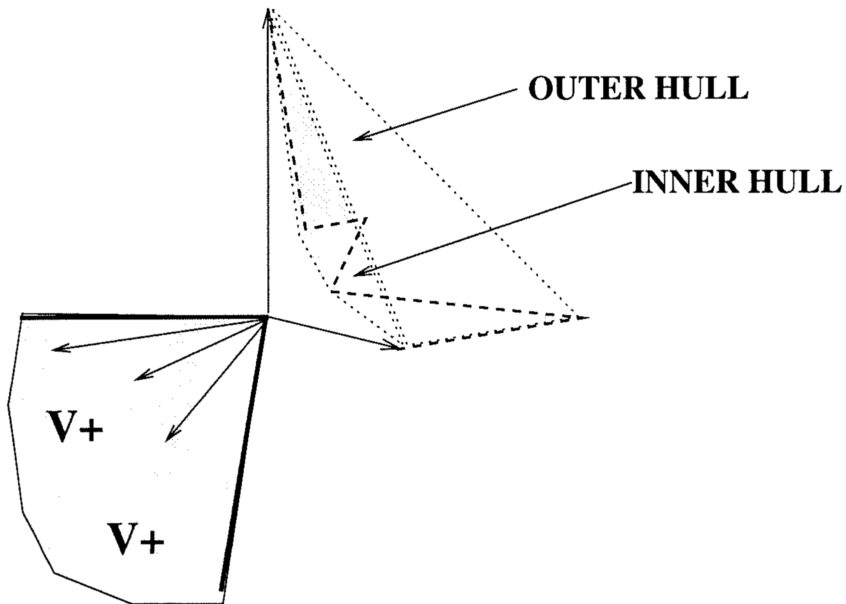


Figure 4.14: Inner and outer hull construction.

Preimages for etching *compound-convex* structures have been called *compensation structures* in the literature. There are many potential topologies for compensation structures. In the space of potential compensation structures there are those with simple vertices, compound vertices, or a mix of the two types. Compensation structures can also have varying numbers of vertices themselves. The primary focus will be to obtain *any* valid preimage. Then later options for choosing from the space of potential solutions will be discussed. Therefore, a compensation structure with three simple vertices will be examined first.

4.3.1 Three Vertex Simple Compensation Structures

A generic three vertex structure and parameters for its complete description are shown in Figure 4.15. To explore the behavior of all possible vertices under anisotropic etching, the parameters ϕ and α_0 were varied in one degree increments to generate the plot in Figure 4.16. ϕ represents the orientation of the vertex relative to the crystal. α_0 is the included angle of the vertex. The white areas of the plot correspond to compound vertices, while the black areas correspond to simple vertices. The etchant used was KOH with principle etch rates, $\langle 310 \rangle = 58 \mu m/hr$, $\langle 110 \rangle = 1 \mu m/hr$, $\langle 100 \rangle = 40 \mu m/hr$. Equation 3.14-3.17 was used to classify each vertex.

To determine which values of the feature angles β_2 and β_3 (see Figure 4.15) correspond to valid compensation structures with three simple vertices, β_2 and β_3 are incremented through their valid ranges and plotted in Figure 4.17 for $\phi = \pi, \alpha_0 = \pi/2$. This is called the $\beta_2\beta_3$ Diagram.

Other criteria may now be used to select from the candidate structures. The boundaries between the valid and invalid regions on the plot represent those compensation structures in which small changes in parameters could result in the simple vertices becoming compound, resulting in undesired etch results. For example, the structure corresponding to $\beta_2 = 30, \beta_3 = 30$, Figure 4.17, may be selected due to its distance from the nearest region boundary. Alternatively, $\beta_2 = 15, \beta_3 = 15$ may be chosen to minimize the size of the preimage, Figure 4.18. (The color bar refers

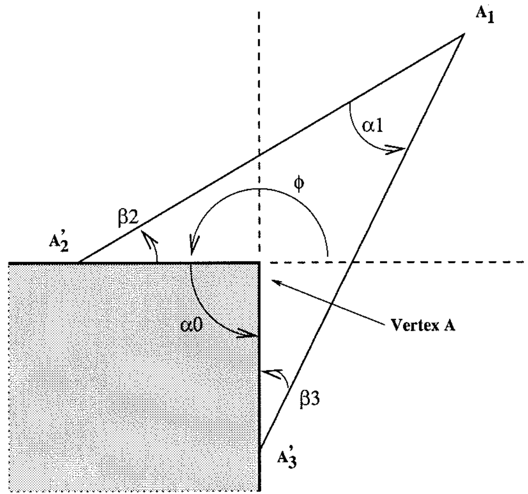


Figure 4.15: Parameterized generic vertex and three point compensation structure.

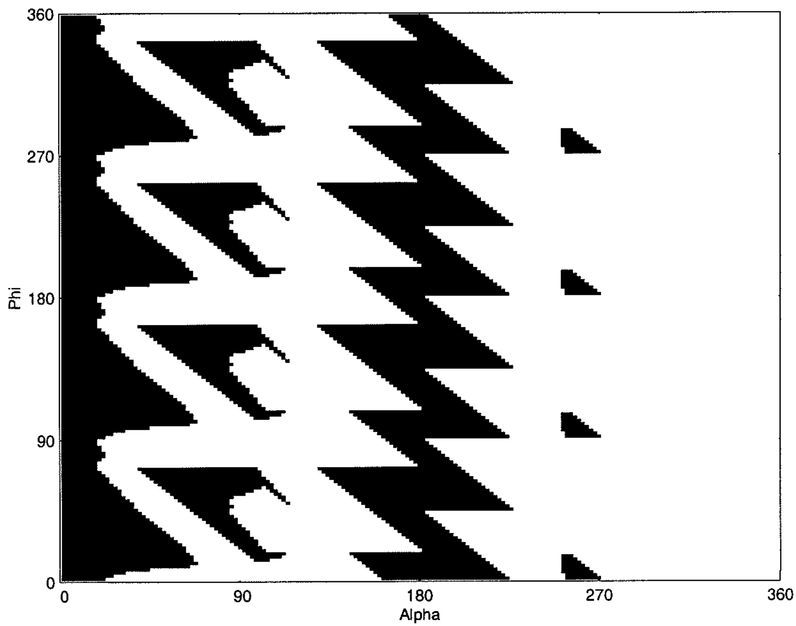


Figure 4.16: $\phi\alpha$ Plot, dark areas are simple vertices.

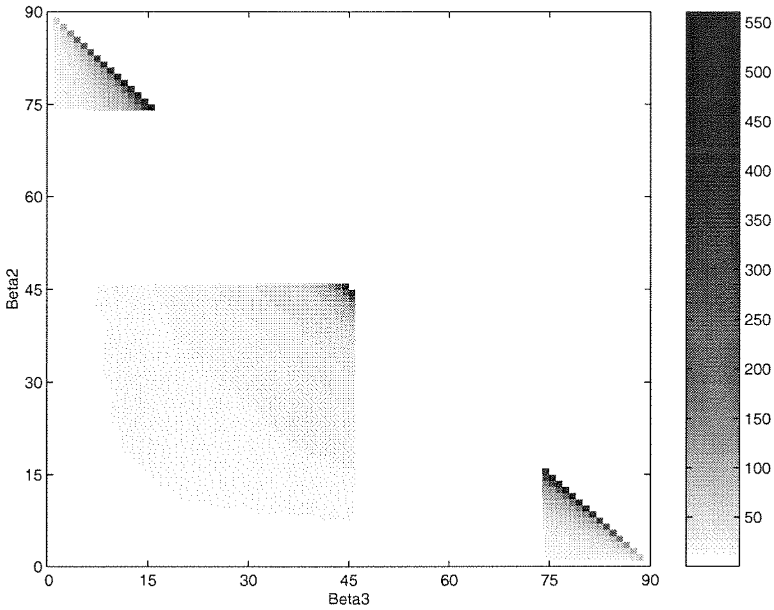


Figure 4.17: $\beta_2\beta_3$ plot, colorbar indicates distance from A to A_1'.

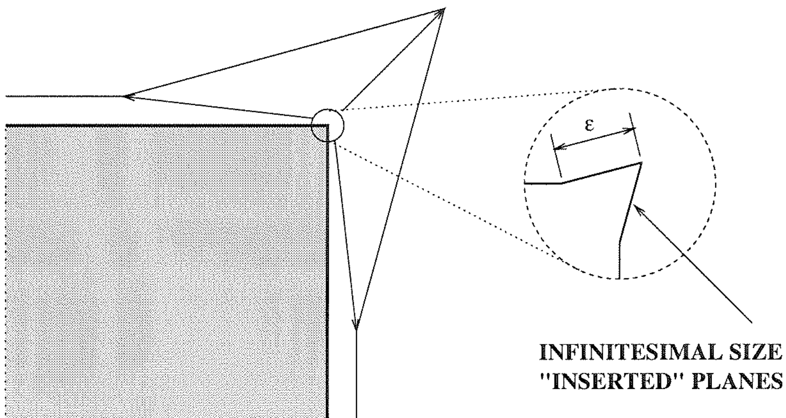


Figure 4.18: Resulting compensation structure.

to the normalized distance from A to A_1'.) Once the structure has been selected, the preimages for its three *simple* vertices may be computed using the methods discussed in the previous sections. Note that the preimages for these vertices are not restricted to *direct* preimages.

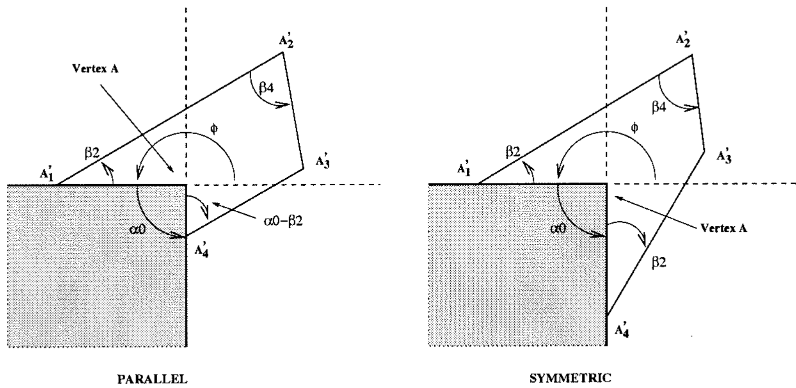


Figure 4.19: Simple four vertex compensation structure parameterizations.

4.3.2 Four Vertex Simple Compensation Structures

In the previous section, compensation structures consisting of three simple vertices were considered. To expand the space of solutions, some compensation structures with four simple vertices are considered. The $\beta_2\beta_3$ diagram was useful for selecting from among the space of potential three vertex solutions. Two subsets of the general four vertex solution are considered that have 2 variable parameterizations, Figure 4.19. Figure 4.20 shows the possible compensation structures for these two parameterizations. Each point in the valid region of the plot corresponds to a local topology alteration at the vertex which results in multiple simple vertices replacing the single compound vertex. A preimage for each of these vertices can then be computed using the methods of § 4.1 and § 4.2.

Note the special case when $\beta_2 = 45^o$ in Figure 4.20. Values of β_2 slightly larger or smaller than this value cause vertices A_1' or A_4' to become compound vertices respectively, Figure 4.19. The slight bump in the plot at $\beta_2 = 82^o, \beta_4 = 18^o$ results from the cusp in the etch rate diagram at 45°. A similar feature is seen on the plot for the symmetric case.

The graph presents a multiplicity of solutions and additional criteria can now be used to select a specific compensation structure. In addition to the criteria of compensation structure "length" used in the previous section, the width, or distance from vertex A to A_1' or A_4', Figure 4.20, is another useful criteria to select from the

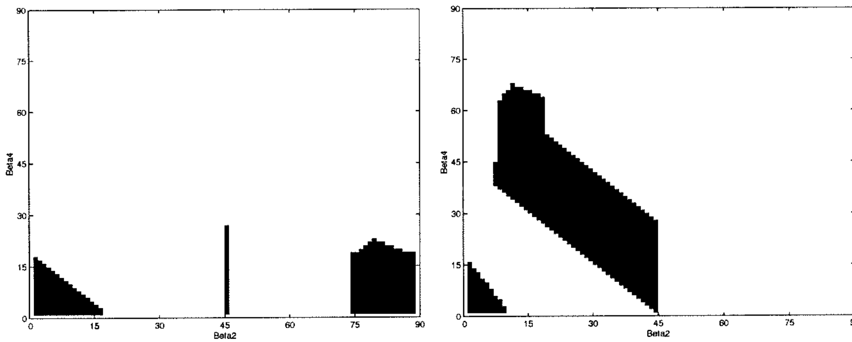


Figure 4.20: $\beta_2\beta_4$ diagrams for four simple vertex compensation structures. Dark regions represent valid structures.

available structures.

This development has been limited to those compensation structures which contain all simple vertices. Alternative approaches will be discussed later in Chapter 5.

4.4 Compound Concave

The final case in our taxonomy of vertex types is the compound concave vertex. Like other compound vertices, there exists no direct preimages. The examples in this presentation have been based on mono-crystalline silicon with etchants whose behavior is similar to potassium-hydroxide (KOH). To date, there have been no empirical or derived mask geometries reported in the literature that will etch a compound-concave vertex. As discussed in § 3.2, the best that can be done for this case is to replace the desired local geometry with an approximation for which a preimage exists.

While the approaches up to this point have been general and applicable to a variety of etchants and crystalline materials, in order to make an approximation, the special structure of the etchant and crystal under consideration are exploited. The general method can be used for other processes, but the details of the approximations and constructions will have to be tailored to the process under consideration.

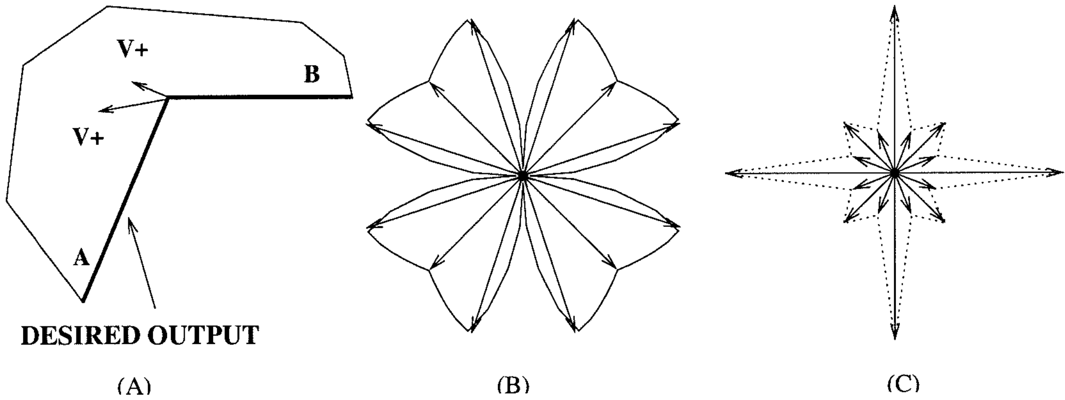


Figure 4.21: (A) Compound concave vertex, (B) Etch rate diagram, (C) Reluctance diagram.

4.4.1 Compound Concave Approximate Preimages

A *compound-concave* vertex is shown in Figure 4.21, formed by planes A and B. While this looks very similar to the simple-concave vertices discussed in § 4.1, note the new vertex contributed to the outer convex hull in Figure 4.22. This indicates that a new plane will appear at this vertex during forward etch and thus it is a compound vertex.

This appearing plane results from the cusp in the etch rate diagram at 45°. This plane is the embodiment of Batterman's comment regarding the dominance of the locally minimum etch rate plane on a concave surface, § 2.4.1.

Locally, any vertex preimage must have the same included angle as the angle between the two planes that form the vertex of the desired feature. Therefore, the plane(s) with locally minimum etch rate that appear at this vertex must appear in the etching of any preimage that includes this angle. Hence, it is not possible to find an exact preimage. Therefore, in the mask synthesis procedure, the desired vertex is replaced with an approximation that includes this appearing plane(s). The size of this appearing plane is determined first.

The approximate solution has a plane of length D, Figure 4.22, which is given by:

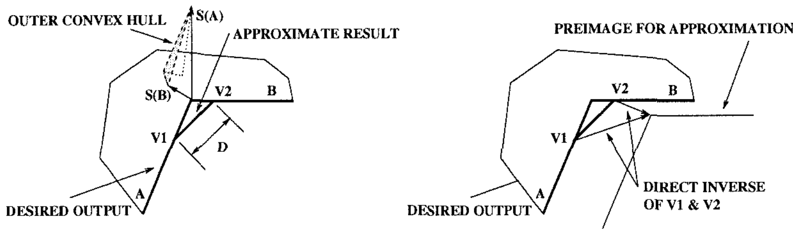


Figure 4.22: Construction of approximation for compound concave vertex.

$$D = (V_1)_t \cdot \tau + (V_2)_t \cdot \tau + \delta \tag{4.4}$$

where τ is the specified etch time, $(V_1)_t$ and $(V_2)_t$ are the tangential velocities of new vertices V_1 and V_2 respectively, and δ is an offset due to the minimum resolution of the mask layout.

This value for D is not necessarily a minimum value. The values of $(V_1)_t$ and $(V_2)_t$ used above assume a direct preimage for each vertex. If the methods of § 4.1.2 and § 4.1.3 are used, a minimum for D may be computed.

Figure 4.23 shows another compound concave vertex, with the value of D in the approximate preimage computed as above. V_1 and V_2 also have feasible indirect preimages (§ 4.1.3). By computing indirect preimages for V_1 and V_2, D is minimized and a preimage is computed that produces a result closer to the desired shape, Figure 4.23.

The availability of the indirect preimages depends strongly on the planes that form the compound concave vertex and the process parameters. Also, the quality of the approximation depends on the extent to which the planes of the indirect preimages can minimize the tangential rate of expansion of the planes in the approximation.

Additionally, mask resolution may be a limiting factor.

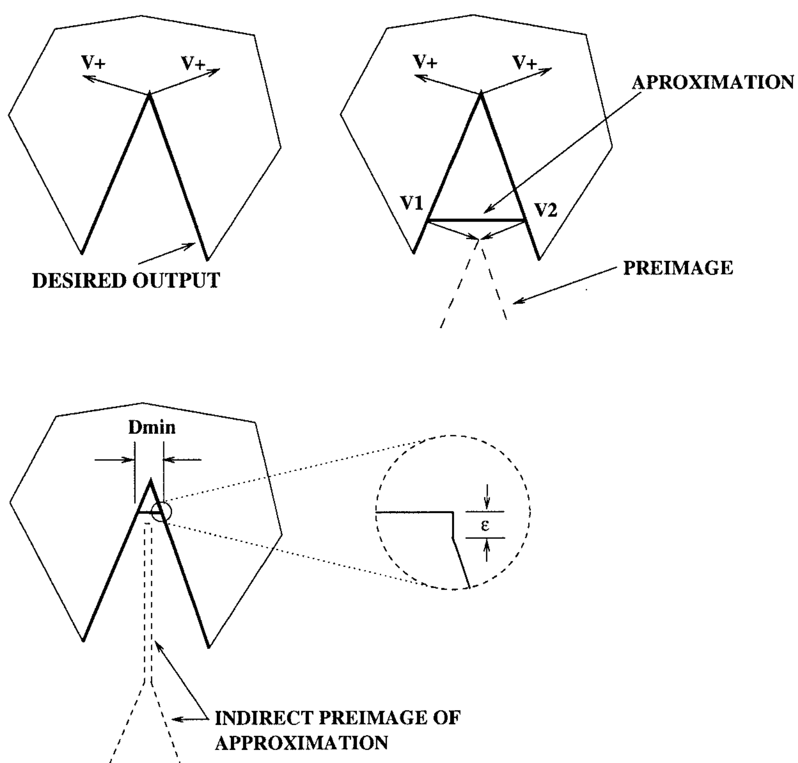


Figure 4.23: Compound concave vertex with improved approximation.

4.5 Example

This section presents an example of using the methods of this chapter to compute the mask layout for the polygon of Figure 4.24. The polygon is arbitrary and developed such that it has at least one vertex of each class. Using the method shown in § 3.1 (Figures 3.4 and 3.5), vertices P_1 and P_2 are determined to be compound convex, vertex P_6 is compound concave, vertices P_5 and P_7 are simple concave, and vertices P_3, P_4, P_8, and P_9 are simple convex.

Vertex Coordinates:

$$
\begin{aligned}
\vec{P}_1 &= (0,500) \\
\vec{P}_2 &= (400,500) \\
\vec{P}_3 &= (400,300) \\
\vec{P}_4 &= (300,0) \\
\vec{P}_5 &= (300,200) \\
\vec{P}_6 &= (200,450) \\
\vec{P}_7 &= (100,200) \\
\vec{P}_8 &= (100,0) \\
\vec{P}_9 &= (0,300)
\end{aligned}
$$

Other Vertex Parameters (Figure 3.6):

$$
\begin{array}{rcl}
\hat{n}_1 = & (1,0) & \hat{\tau}_1 = (0,1) \\
\hat{n}_2 = & (0,-1) & \hat{\tau}_2 = (1,0) \\
\hat{n}_3 = & (-1,0) & \hat{\tau}_3 = (0,-1) \\
\hat{n}_4 = & (-0.949,0.316) & \hat{\tau}_4 = (-0.316,-0.949) \\
\hat{n}_5 = & (1,0) & \hat{\tau}_5 = (0,1) \\
\hat{n}_6 = & (0.949,0.316) & \hat{\tau}_6 = (-0.316,0.949) \\
\hat{n}_7 = & (-0.949,0.316) & \hat{\tau}_7 = (-0.316,-0.949)
\end{array}
$$

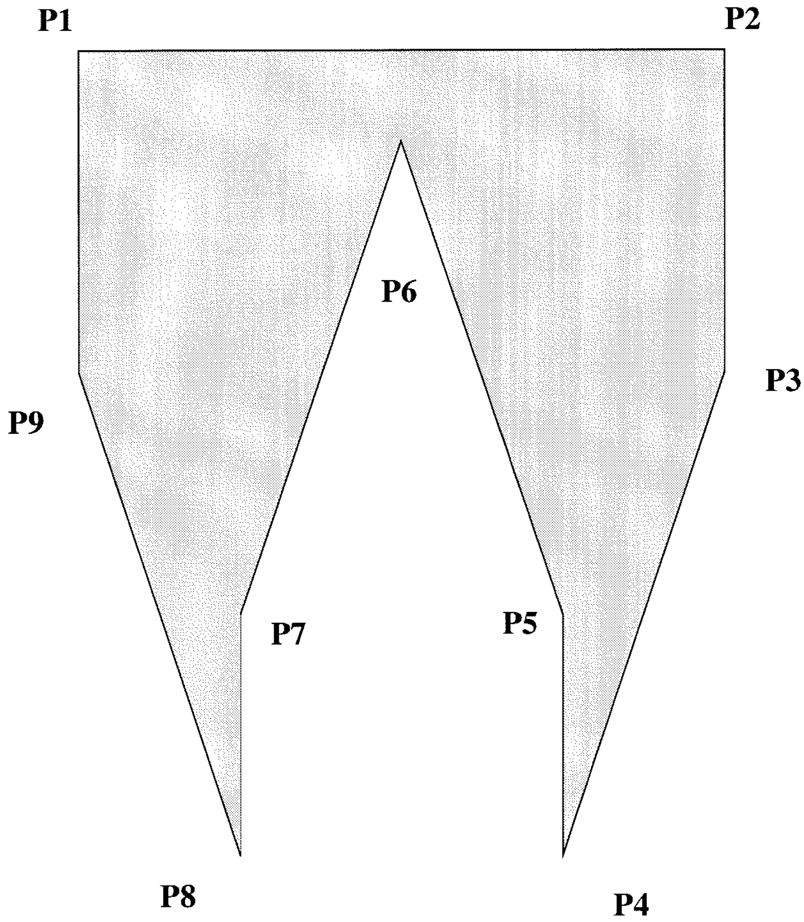


Figure 4.24: Desired shape for mask synthesis example.

$$\hat{n}_8 = (-1, 0) \qquad \hat{\tau}_8 = (0, -1)$$

$$\hat{n}_9 = (0.949, 0.316) \qquad \hat{\tau}_9 = (-0.316, 0.949)$$

Etchant Parameters, Figure 4.25:

$$\langle 310 \rangle = 50\mu m/hr$$

$$\langle 100 \rangle = 40\mu m/hr$$

$$\langle 110 \rangle = 1\mu m/hr$$

$$\text{Desired Etch Time} = 1 \text{ hour}$$

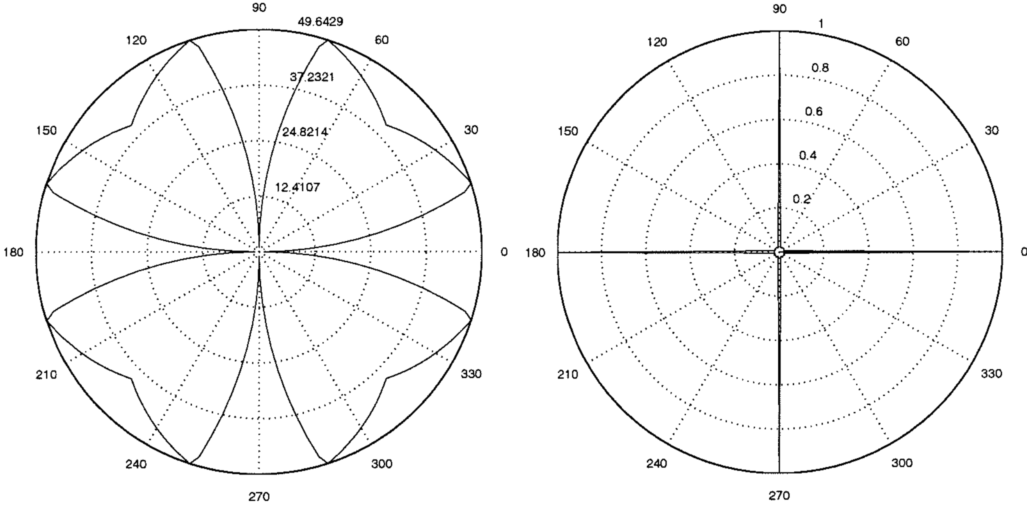


Figure 4.25: Etch rate and etch reluctance diagrams.

4.5.1 Compound Convex Vertices

Preimages for vertices P_1 and P_2 are computed using the method of § 4.3.1. Figure 4.17 applies to both of these vertices and values of $\beta_2 = \beta_3 = 18.43$ is chosen for the three vertex compensation structure. This value is chosen for its relatively small size of compensation structure, Figure 4.17, and because it corresponds to the $\langle 310 \rangle$ direction, which is a dominant plane for the KOH etchant to be used. Once the values of β_2 and β_3 are chosen, P_1 and P_2 are replaced with an infinitesimal local topology, Figures 4.18 and 4.26. These six new simple vertices replace the two compound vertices and will be considered below with the other simple vertices.

New Vertex Data:

$$
\begin{array}{rcl}
lll\hat{n}_{1,1} = & (1,0), & \hat{\tau}_{1,1} = (0,1) \\
\hat{n}_{1,2} = & (0.949, 0.316), & \hat{\tau}_{1,2} = (-0.316, 0.949) \\
\hat{n}_{1,3} = & (-0.316, -0.949), & \hat{\tau}_{1,3} = (0.949, -0.316) \\
\hat{n}_{2,1} = & (0,-1), & \hat{\tau}_{2,1} = (1,0) \\
\hat{n}_{2,2} = & (0.316, -0.949), & \hat{\tau}_{2,2} = (0.949, 0.316)
\end{array}
$$

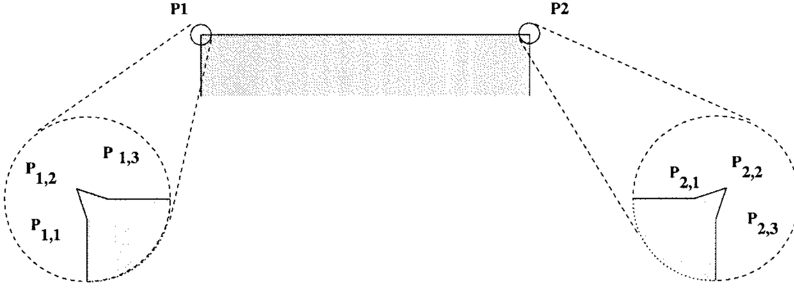


Figure 4.26: Compound convex vertices decomposed into multiple simple vertices.

$$\hat{n}_{2,3} = \quad (-0.316, 0.949), \quad \hat{\tau}_{2,3} = (-0.949, -0.316)$$

4.5.2 Compound Concave Vertex

The preimage for the compound concave vertex, P_6, will be computed using the method of § 4.4. Based on the convex hull analysis, it is apparent that a $\langle 110 \rangle$ crystal plane will appear at vertex P_6 or any topologically similar vertex. Figure 4.27 shows an approximated vertex defined by $P_{6,1}$ and $P_{6,2}$. The vertical location of this new line segment in the mask layout and thus the coordinates of the new vertices will depend on the minimum size of the approximation, Equation 4.4.

Vertex Data:

$$\hat{n}_{6,1} = \quad (0.949, 0.316), \quad \hat{\tau}_{6,1} = (-0.316, 0.949)$$

$$\hat{n}_{6,2} = \quad (0, 1), \quad \hat{\tau}_{6,2} = (-1, 0)$$

Vertices $P_{6,1}$ and $P_{6,2}$ are simple concave vertices and can be solved using the indirect preimage technique of § 4.1.3 to minimize the component of the vertex velocity vector tangent to the plane, Figures 4.7 and 4.8. The indirect preimage technique involves inserting a new vertical plane of infinitesimal size forming another

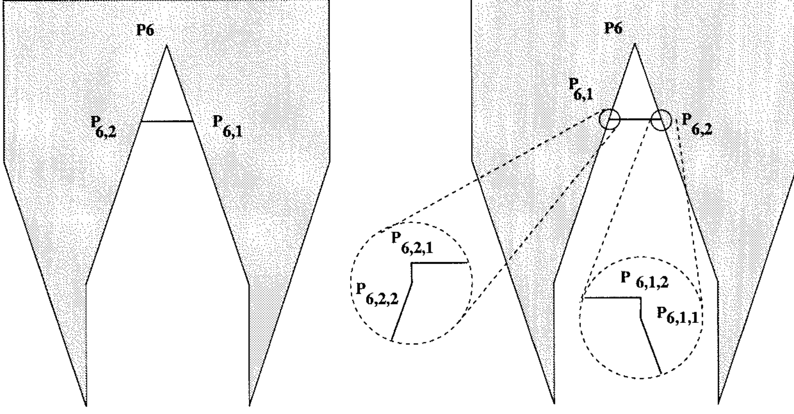


Figure 4.27: Compound concave vertex with approximation.

pair of vertices.

Vertex Data:

$$\hat{n}_{6,1,1} = (0.949, 0.316), \quad \hat{\tau}_{6,1,1} = (-0.316, 0.949)$$
$$\hat{n}_{6,1,2} = (1, 0), \quad \hat{\tau}_{6,1,2} = (0, 1)$$
$$\hat{n}_{6,2,1} = (0, 1), \quad \hat{\tau}_{6,2,1} = (-1, 0)$$
$$\hat{n}_{6,2,2} = (-1, 0), \quad \hat{\tau}_{6,2,2} = (0, -1)$$

To determine the coordinates for the vertices, the tangential component of the velocity of vertices $P_{6,1,2}$ and $P_{6,2,1}$ is computed from Equation 3.5 and entered into Equation 4.4.

$$V_i = r_i \cdot \hat{n}_i + [\frac{r_{i+1} - r_i \hat{n}_i \cdot \hat{n}_{i+1}}{\hat{\tau}_i \cdot \hat{n}_{i+1}}] \cdot \hat{\tau}_i$$

$$\vec{V}_{6,1,2} = r_{\langle 110 \rangle} \hat{n}_{6,1,2} + [\frac{r_{\langle 110 \rangle} - r_{\langle 110 \rangle} \hat{n}_{6,1,2} \cdot \hat{n}_{6,2,1}}{\hat{\tau}_{6,1,2} \cdot \hat{n}_{6,2,1}}] \cdot \hat{\tau}_{6,1,2} \quad (4.5)$$
$$= (1, 1) \mu m/hr$$

$$\vec{V}_{6,2,1} = r_{\langle 110 \rangle} \hat{n}_{6,2,1} + [\frac{r_{\langle 110 \rangle} - r_{\langle 110 \rangle} \hat{n}_{6,2,1} \cdot \hat{n}_{6,2,2}}{\hat{\tau}_{6,2,1} \cdot \hat{n}_{6,2,2}}] \cdot \hat{\tau}_{6,2,1} \quad (4.6)$$
$$= (-1, 1) \mu m/hr$$

$$\begin{aligned} D &= \vec{V}_{6,1,2} \cdot (1,0) * 1\text{hour} + \vec{V}_{6,2,1} \cdot (-1,0) * 1\text{hour} \\ &= 2.0\mu m \end{aligned} \quad (4.7)$$

Using the value for D and the geometry of P_5, P_6, and P_7:

$$P_{6,1,1} = (201, 447)$$
$$P_{6,1,2} = (201, 447)$$
$$P_{6,2,1} = (199, 447)$$
$$P_{6,2,2} = (199, 447)$$

The location coordinates and values for $\hat{n}$ and $\hat{\tau}$ are now known for the four simple vertices that approximate the original compound concave vertex. The direct preimages will be solved below with the other simple vertices.

4.5.3 Simple Vertices

The vertices in the polygon of Figure 4.24 have now been replaced with a set of vertices that have direct preimages. What remains is to compute the vertex velocity vectors for each vertex and simulate in "reverse time." Using the data for the location, normal, and tangent for each vertex given above with Equation 3.5, the vertex velocity vectors are:

$$\begin{aligned} \vec{V}_{1,1} &= (1.0000, 155.2247) \\ \vec{V}_{1,2} &= (79.0160, -78.9980) \\ \vec{V}_{1,3} &= (-155.2970, -1.0000) \\ \vec{V}_{2,1} &= (155.2247, -1.0000) \\ \vec{V}_{2,2} &= (-78.9980, -79.0160) \\ \vec{V}_{2,3} &= (-1.0000, 155.2970) \end{aligned}$$

$$\vec{V}_3 = (-1.0000, 155.2247)$$
$$\vec{V}_4 = (1.0000, 161.3033)$$
$$\vec{V}_5 = (1.0000, 155.2247)$$
$$\vec{V}_{6,1,1} = (1.0000, 155.2970)$$
$$\vec{V}_{6,1,2} = (1.0000, 1.0000)$$
$$\vec{V}_{6,2,1} = (-1.0000, 1.0000)$$
$$\vec{V}_{6,2,2} = (-1.0000, 155.2247)$$
$$\vec{V}_7 = (-1.0000, 155.2970)$$
$$\vec{V}_8 = (-1.0000, 161.2310)$$
$$\vec{V}_9 = (1.0000, 155.2970)$$

With a desired etch time of one hour, the mask layout for the shape of Figure 4.24 is shown in Figure 4.28. For verification, this mask is simulated using the SEGS simulator [23] and the results are shown in Figure 4.29. While experimental validation was not available, the compensation structures for P_1 and P_2 have been experimentally validated in [64], Figure 4.30.

$$\vec{P}_{1,1} = (-1.0000, 344.7753)$$
$$\vec{P}_{1,2} = (-79.0160, 578.9980)$$
$$\vec{P}_{1,3} = (155.2970, 501.0000)$$
$$\vec{P}_{2,1} = (244.7753, 501.0000)$$
$$\vec{P}_{2,2} = (478.9980, 579.0160)$$
$$\vec{P}_{2,3} = (401.0000, 344.7030)$$
$$\vec{P}_3 = (401.0000, 144.7753)$$
$$\vec{P}_4 = (299.0000, -161.3033)$$
$$\vec{P}_5 = (299.0000, 44.7753)$$
$$\vec{P}_{6,1,1} = (200.0000, 291.7030)$$

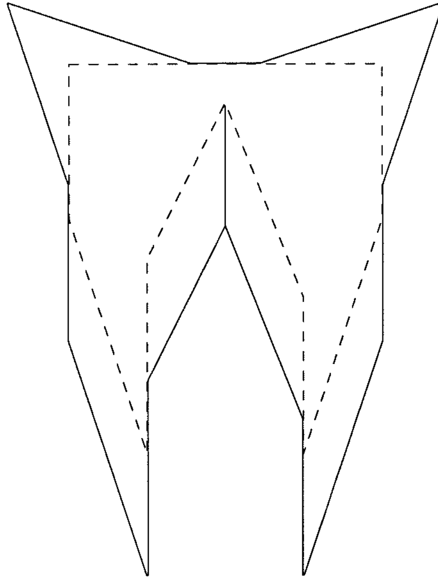


Figure 4.28: Final mask layout (solid line) with original shape (dashed line).

$$\begin{aligned}
\vec{P}_{6,1,2} &= (200.0000, 446.0000) \\
\vec{P}_{6,2,1} &= (200.0000, 446.0000) \\
\vec{P}_{6,2,2} &= (200.0000, 291.7753) \\
\vec{P}_{7} &= (101.0000, 94.7030) \\
\vec{P}_{8} &= (101.0000, -161.2310) \\
\vec{P}_{9} &= (-1.0000, 144.7030)
\end{aligned}$$

4.6 Robustness

The development in the previous sections have assumed an accurately timed etch with a well characterized and uniform etch process. This section will look at how the options made available by the indirect preimage can be used to account for some variations in the etch time or the characteristic parameters of the etch process.

For example, variations in the concentration of the etchant can cause "loading effects" where the etch parameters are non-uniform across the surface of the wafer.

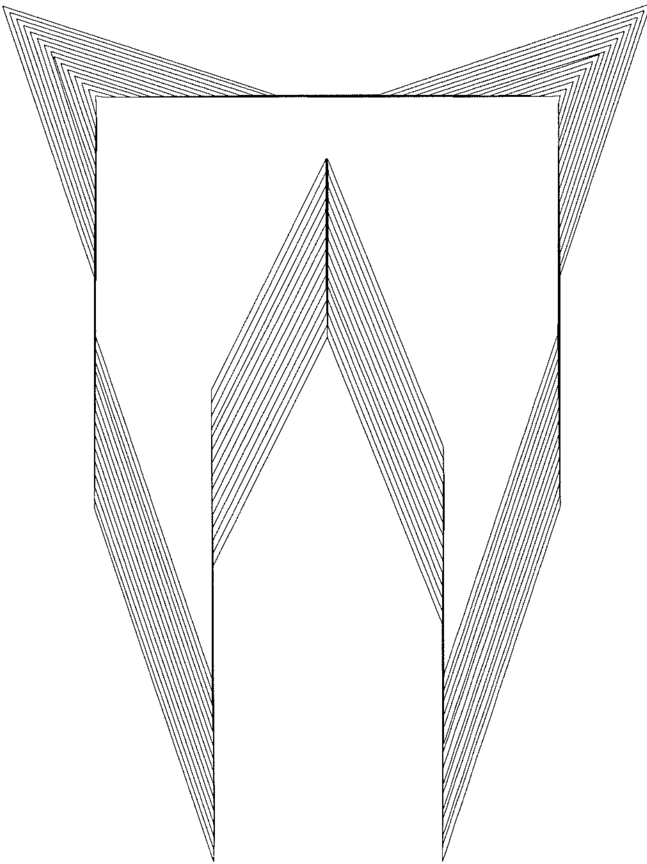


Figure 4.29: Simulation results for example mask layout. Outer shape is the computed mask, equal etch time steps to final etched shape.

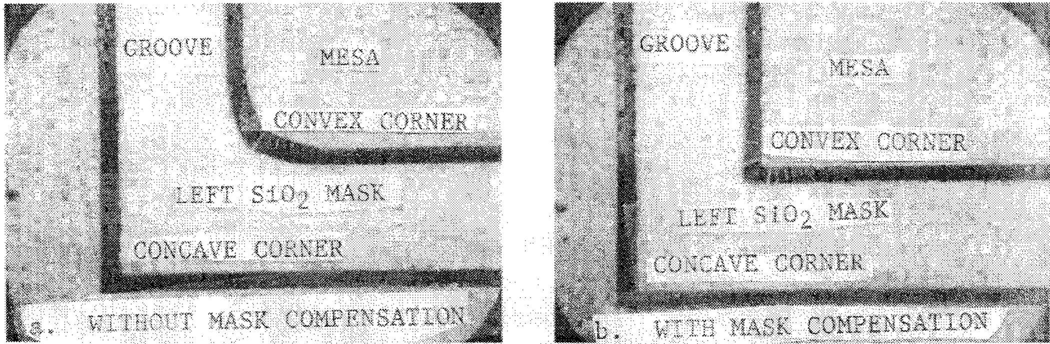


Figure 4.30: Experimental validation of corner compensation [64].

This can cause the etch rate diagram to be scaled and result in under- or over-etching locally. If the vertex velocity vectors can be minimized at the end of the etch, the impact of the variations in etch time can be reduced.

In § 4.3.1, the $\beta_2\beta_3$ diagram (Figure 4.17) was parameterized by the size of the compensation structure. Figure 4.31, shows a new $\beta_2\beta_3$ diagram plotted with the grey-scale value corresponding to the maximum vertex velocity vector in the compensation structure. Therefore, by choosing $\beta_2, \beta_3 \in [20^o, 30^o]$, errors can be reduced.

Figure 4.32 shows two compensation structures under-etched and over-etched. The first used the $\beta_2 = \beta_3 = 20^o$, the second used $\beta_2 = \beta_3 = 40^o$. Due to the decreased vertex velocity in compensation structure "A", it is less effected by the under-etch. Note that both over-etch similarly, this is because both compensation structures will etch to the desired vertex at the specified time and will proceed identically during the over-etch period.

Similarly for simple preimages, Figure 4.33 shows the results of under-etching the direct and indirect preimages developed in § 4.2. Note that while the local topology of the under-etched direct preimage is identical to the desired result it has a great spatial deviation from the desired. Because the indirect preimage has reduced vertex velocities, it is closer to the desired result.

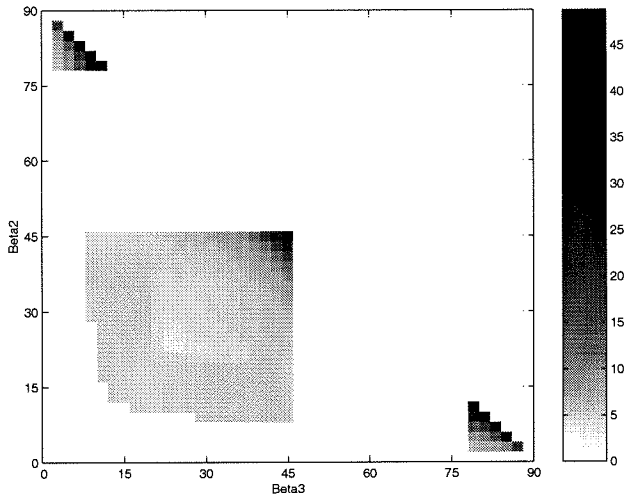


Figure 4.31: $\beta_2\beta_3$ diagram, greyscale refers to maximum vertex velocity.

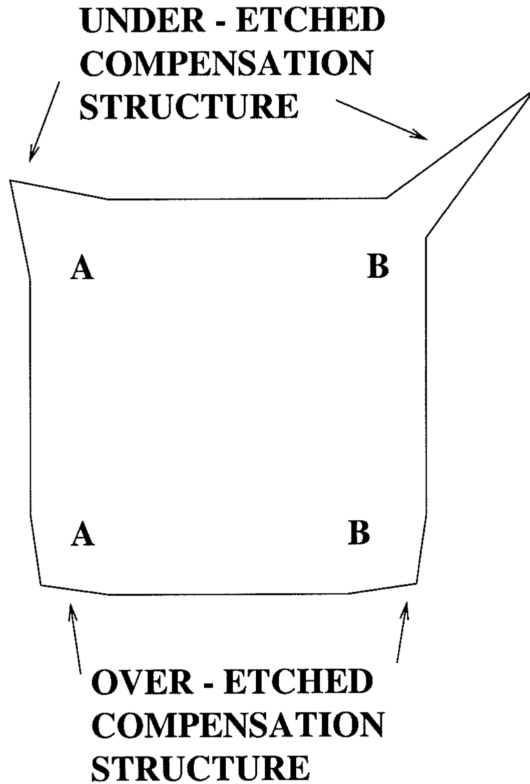


Figure 4.32: Under- and Over- etched compensation strucutres.

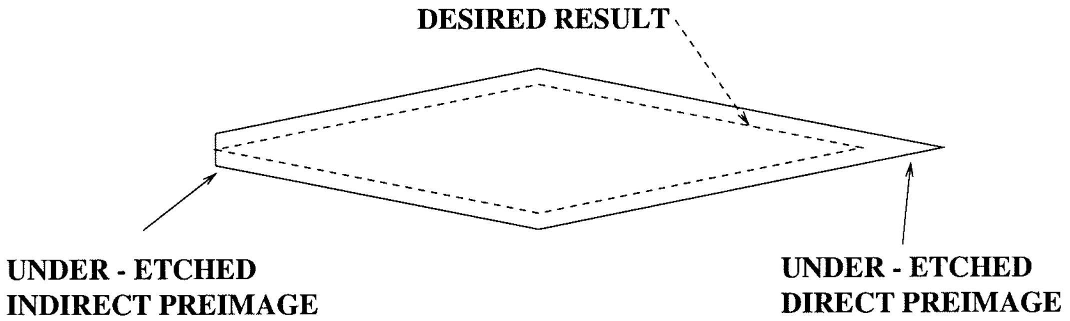


Figure 4.33: Under-etched simple convex preimages.

4.7 Summary

This chapter has presented methods to compute the preimage for each of the four classes of vertices, simple-concave, simple-convex, compound-concave, and compound-convex. Where available, direct preimage constructions were demonstrated. As alternative solutions, and for the cases in which no direct preimages where available, methods where developed to compute indirect preimages with simple vertices. The next chapter extends this development to preimages with compound vertices.

Chapter 5

Timing Models for Mask Synthesis

The results in the previous chapter relied on reducing complex features to multiple simple vertices and computing the direct inverse for each one. The result was, therefore, a preimage with all simple vertices. There are examples in the literature of single purpose mask layouts and compensation structures that have compound vertices. This section discusses extending the techniques in this thesis to compute preimages that contain compound vertices.

The approach was hinted at in § 4.4.1, Figure 4.22. During the reverse simulation of vertex velocity vectors ($V1, V2$, Figure 5.1), two direct preimage inverses meet at a point A. i.e., A is a compound vertex whose appearing plane is bounded by V_1, V_2. An indirect preimage for vertex A can be computed and the preimage calculation continued backward in time. If the appearing plane at the compound vertex is different, then the preimage computation must be halted before the vertices meet, or an alternative preimage must be computed for one or both of the original vertices. This condition arises from the fact that if backward-in-time calculations are continued when the vanishing plane is not the appearing plane, the forward evolution of the etch will be different from the backward evolution taken during the synthesis process and the desired geometry will not be realized.

Since appearing planes are planes with locally minimum or maximum etch rates, if the modifications vertices local geometry are limited to these dominant planes, the likelihood that preimage calculations can be continued through these merge points

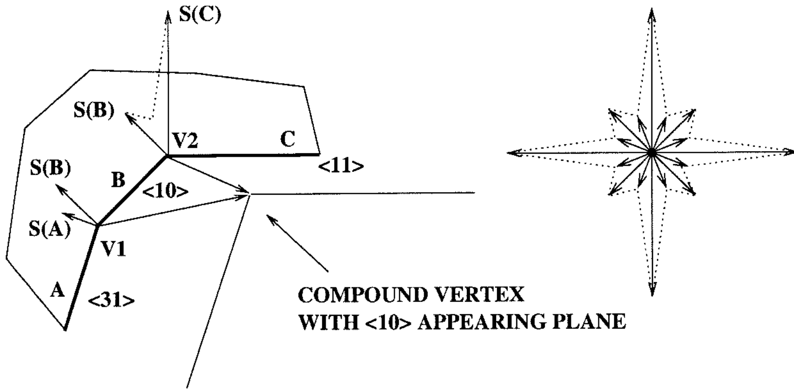


Figure 5.1: Vertex merging during preimage construction.

is increased.

Figure 5.2(a) shows a simple concave vertex with the preimage constructed in the usual way. In previous sections it was suggested that a preimage which minimized the rate of vertex travel tangent to "plane A" during etch would be desired in order to reduce the potential conflict with neighboring preimages or to improve the approximation for compound concave vertices.

In Figure 5.2(b), the topology is modified with two dominant planes, C and D (⟨31⟩ and ⟨13⟩ respectively) inserted at a distance l_1 (Figure 5.2) from the vertex. Note, as discussed in § 4.3, when the feature topology is modified with planes of infinitesimal size which form simple vertices, the desired output will be recovered in the forward etch. Using the timing equations (§ 3.1.1, equations 3.11, 5.1), time $-T_1$ is related to l_1 as follows.

$$T_1 = l_1 \cdot (-G_1)^{-1} \tag{5.1}$$

where G_1, Equation 3.8, is the growth rate of the plane between V_1 and V_2. As the planes evolve in *reverse* time, the preimage for vertex V_2 will meet the preimage for V_1 at time $-T_1$. The new preimage vertex formed at $-T_1$ is a compound concave vertex, V_5. The appearing plane at V_5 arises from the same reluctance diagram

as the plane that was bounded by V_1 and V_2. If a preimage for V_5 can be found, the preimage calculation can continue in reverse time beyond $-T_1$. A preimage for the vertex V_5 can be computed using the alternative indirect solution shown in Figure 4.8, § 4.1.3. This alternative indirect solution introduces a new plane to the topology at V_5, which forms V_6 and V_7. During time $0 \rightarrow -T_1$, plane C has grown to length l_2 given by:

$$l_2 = T_1 \cdot G_C \tag{5.2}$$

The evolution of the new vertex V_7 during reverse simulation will decrease l_2, and at time $-T_2$, $l_2 = 0$.

$$T_2 = l_2 \cdot (-G'_C)^{-1} \tag{5.3}$$

where $G'_C < 0$ is the growth rate for plane C during the second phase ($T_1 < t < T_2$) of preimage construction. Since the neighboring planes of C during T_1 and T_2 are different, $G_C \neq G'_C$. At time T_2, V_7 and V_8 meet to form a compound convex vertex. Given a desired etch time, τ, the new preimage solution can now be solved.

$$\begin{aligned}
\tau &= T_1 + T_2 \\
&= T_1 + l_2 \cdot (-G'_C)^{-1} \\
&= T_1 \cdot [1 + G_C \cdot (G'_C)^{-1}] \\
T_1 &= \tau \cdot [1 + G_C \cdot (G'_C)^{-1}]^{-1} && (5.4) \\
& && (5.5)
\end{aligned}$$

then,

$$l_1 = T_1 \cdot (-G_1)$$

$$= -\frac{\tau \cdot G_1}{1 + G_C \cdot (G'_C)^{-1}} \text{ where } G_1, G'_C < 0 \qquad (5.6)$$

Note that the compound convex vertex formed by V_6 and V_7 does have available preimages discussed in § 4.3, and this process could be continued. The system of equations 5.1- 5.6 would increase in order, but remain solvable, as long as valid local preimages and physical space on the wafer allowed.

This timing model approach allows many of the single purpose compensation structures reported in the literature to be replicated. However, since only the general behavior of KOH on mono-crystalline Si was used, the method is sufficiently general to allow computation of preimages for a variety of etchants, temperatures and solution concentrations.

5.1 Compensation Structure Example

The compensation structures in the previous chapter contained 3 or 4 vertices that were all of type *simple*. This section explores methods to derive compensation structures that may have *compound* vertices. By definition, a *compound* vertex exhibits appearing planes as the etch begins. Therefore, a *compound* vertex formed by a synthesis method must have those appearing planes disappear exactly at the end of the "reverse simulation." As discussed in § 3.2, the topological evolution of the "reverse simulation" can be altered by inserting appropriate infinitesimal planes during the synthesis procedure.

The timing model proposed in this chapter parameterizes these topology bifurcations and encodes the shape of a compensation structure in a system of equations. Once the precise orientation of the geometry and etch process parameters are known, the equations are solved to determine the time and location of the topology changes in order that *compound* vertices are formed appropriately.

In [34, 65], compensation structures are discussed for use in narrow channels, Figure 5.4. While a formulation is provided to modify the structure for different etch rates, the guidelines are very application specific.

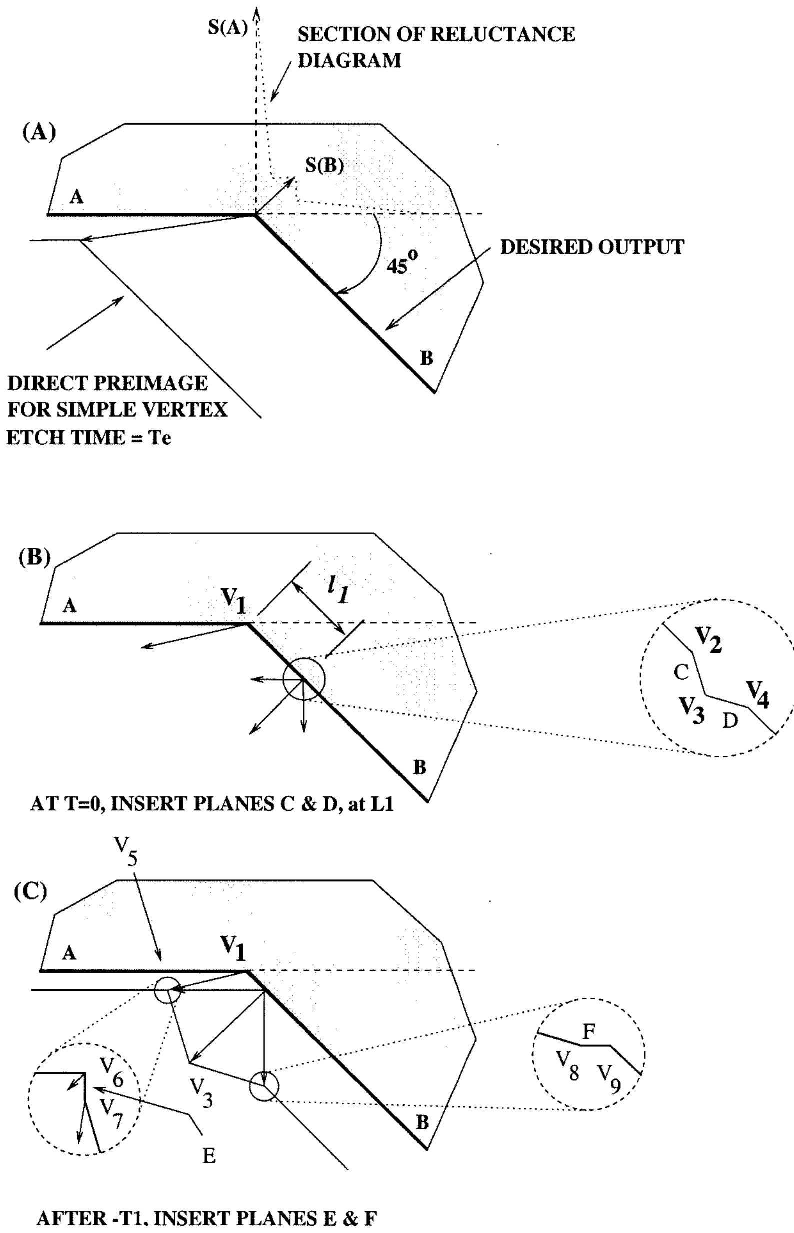


Figure 5.2: Preimage construction with timing model.

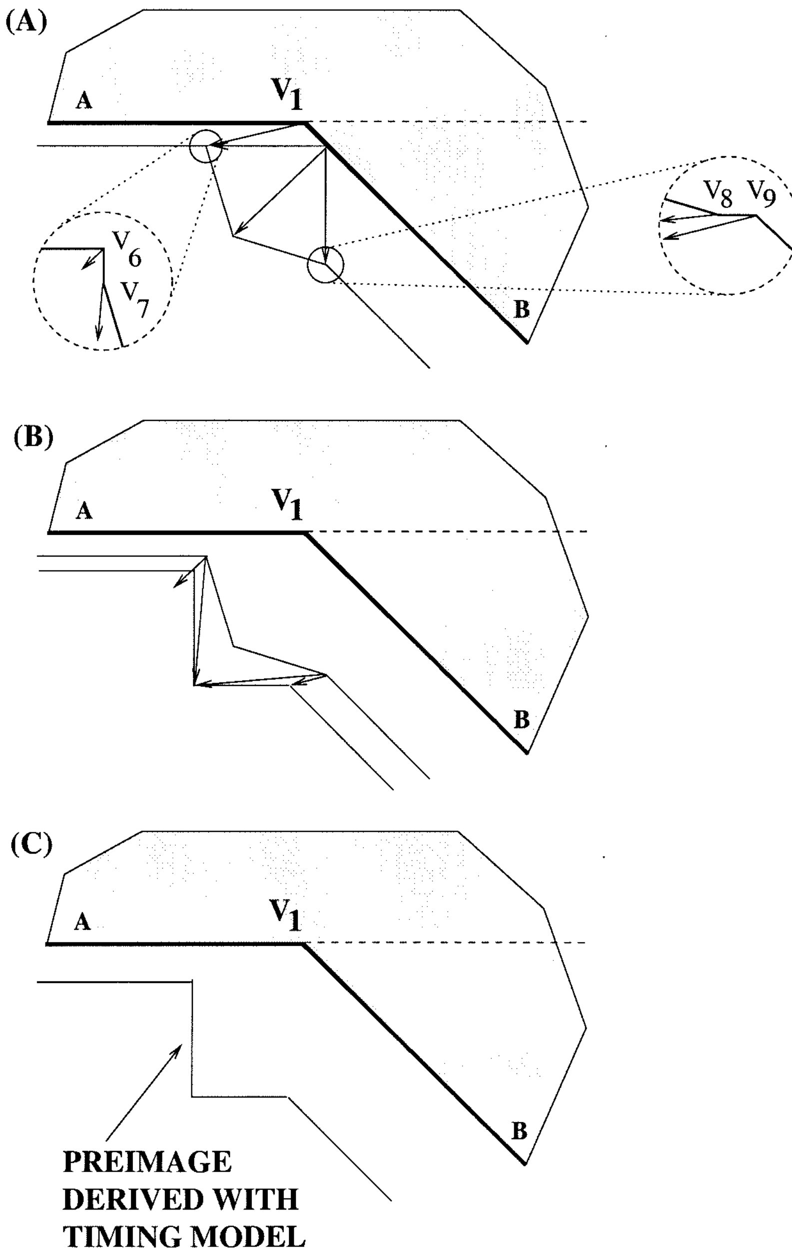


Figure 5.3: Preimage construction with timing model.

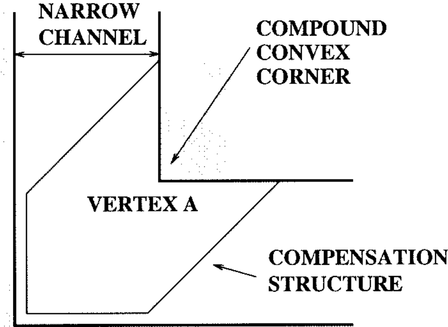


Figure 5.4: Compensation structure for vertex A, after [65].

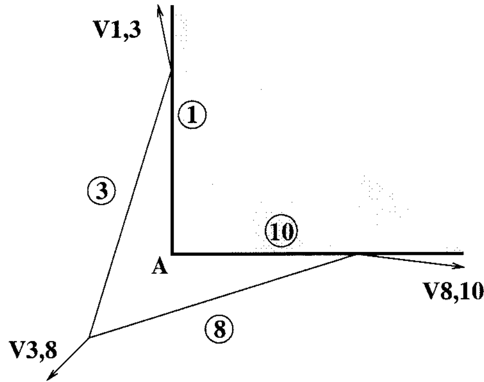


Figure 5.5: Basic 3 point compensation structure.

What follows encodes the compensation structure found in [65] in a general formulation that can be optimized based on the designers needs. Further, the process demonstrates how any compensation structure can be generalized.

First the method of the previous chapter is used to choose a starting point. Figure 4.17 shows the selection of simple 3 point structures available for this vertex. Since the timing model relies on topology modification and ultimately on the structure of appearing and disappearing planes, no planes can be added to the reverse simulation that will not reappear during etching. Therefore, the selection of points from Figure 4.17 is restricted to the dominant etch planes. Here, the $\langle 310 \rangle$ planes are chosen and inserted into the topology, as seen in Figure 5.5. Advancing in re-

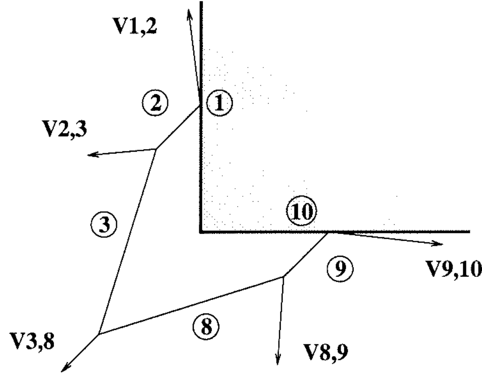


Figure 5.6: Insert ⟨100⟩ planes at time, $-t_1$.

verse simulation time, ⟨100⟩ planes are inserted at time t_1, Figure 5.6 (planes 2 and 9).

Similarly at time $-(t_1 + t_2)$, ⟨310⟩ planes 5 and 6 are inserted in the topology, Figure 5.7. Note that each of these planes is selected because of its local dominance in the etch rate diagram. Therefore, in reverse time the inserted plane will grow, but during the forward etch the plane will shrink, disappearing at time, $-(t_1 + t_2)$.

The last step is to insert ⟨110⟩ planes at time $-(t_1 + t_2 + t_3)$, Figure 5.8. The diagram shows that as the reverse etch proceeds for an additional time t_4, $V_{4,5}$, $V_{5,6}$, and $V_{6,7}$ intersect to form the *compound* corner of the compensation structure. For the method to be valid, the total desired etch time, T, must equal the sum of the increments, $t_1 + t_2 + t_3 + t_4$.

To solve for each of the intermediate times, each of the vertex velocities is first derived, Equation 3.5:

$$V_i = r_i \cdot \hat{n}_i + [\frac{r_{i+1} - r_i \hat{n}_i \cdot \hat{n}_{i+1}}{\hat{\tau}_i \cdot \hat{n}_{i+1}}] \cdot \hat{\tau}_i \tag{5.7}$$

where r_i, $\hat{n}_i$, and $\hat{\tau}_i$ are the etch rate, normal, and tangent unit vectors of the appearing plane at vertex i.

The position of each vertex during the etch evolution is expressed as:

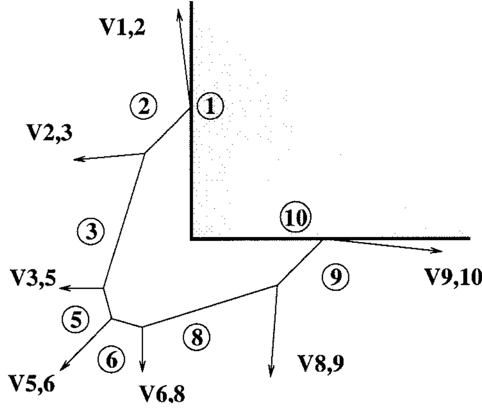


Figure 5.7: Insert $\langle 310 \rangle$ planes at time, $-t_2$.

$$P_5 = V_{3,8}\, t_1 + V_{3,8}\, t_2 + V_{5,6}\, t_3 + V_{5,6}\, t_4 \quad (5.8)$$

$$P_6 = V_{3,8}\, t_1 + V_{3,8}\, t_2 + V_{6,8}\, t_3 + V_{6,7}\, t_4 \quad (5.9)$$

$$P_7 = V_{3,8}\, t_1 + V_{3,8}\, t_2 + V_{6,8}\, t_3 + V_{7,8}\, t_4 \quad (5.10)$$

$$P_8 = V_{8,10}\, t_1 + V_{8,9}\, t_2 + V_{8,9}\, t_3 + V_{8,9}\, t_4 \quad (5.11)$$

$$T = t_1 + t_2 + t_3 + t_4 \quad (5.12)$$

where P_k is the position of vertex k, $V_{i,j}$ is the velocity of the vertex formed by planes i and j, and t_n is the etch time, Figure 5.4 - 5.8.

For this example, the compensation structure is symmetric; therefore, only vertices P_5 through P_8 are examined. The relation to determine the etch time are given by: $P_5 = P_6$ (plane 6 disappearing) and $P_7 = P_8$ (plane 8 disappearing). Note that since planes 6 and 8 are locally dominate planes for convex vertices, they will appear immediately as the etch commences. From Equations 5.8 and 5.9:

$$P_5 = P_6 \text{ yields: } t_3 = \alpha\, t_4 \quad (5.13)$$

$$\alpha = \left|\left|\frac{V_{6,7} - V_{5,6}}{V_{5,6} - V_{6,8}}\right|\right| \quad (5.14)$$

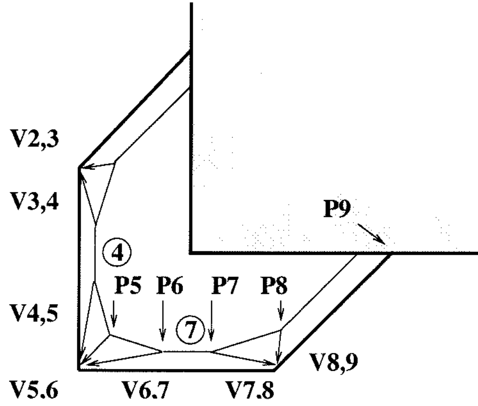


Figure 5.8: Insert $\langle 110 \rangle$ planes at time, $-t_3$.

where α is a function of the etch rates. Equations 5.13 and 5.12 yield:

$$t_4 = \frac{T - t_1 - t_2}{1 + \alpha} \tag{5.15}$$

The second relation ($P_7 = P_8$) with Equations 5.12 gives:

$$[A] \begin{pmatrix} t_1 \\ t_2 \end{pmatrix} = \vec{B}(T) \tag{5.16}$$

Up to this point no information about the etch rates or process parameters have been included in the equations. Once the geometric structure has been encoded by the above equations, it can be solved on a case by case bases for specific process parameters.

To solve the system, etch rates are specified. For KOH, $\langle 110 \rangle = 1\mu m/\text{hr}$, Rate $\langle 310 \rangle = 60\mu m/\text{hr}$, and Rate $\langle 100 \rangle = 50\mu m/\text{hr}$.

Due to the symmetries involved, matrix $[A]$ is singular which allows another degree of freedom in the specification. Since the channel width is to be minimized, $\min(P_8 \cdot \hat{y})$ will be used to resolve the extra parameter. Equation 5.11 can be rewritten using Equations 5.12 and 5.13 as $P_8(t_1, t_2)$. The minimum corresponds to $t_2 = 0$, thus planes 2,5,6, and 9 are inserted at the same time. Equation 5.16 can then be solved for t_1 as a function of the etch Time T. $t_1 = 0.1856\ T$, $t_2 = 0$,

$t_3 = 0.6076\ T$,$t_4 = 0.2069\ T$.

5.1.1 SIMULATION

The above compensation structure was experimentally verified in [65], Figure 5.10, and a simulation of the etching using the SEGS simulator (http://micron.me.tuns.ca) [23, 24] is shown in Figure 5.9.

The synthesis method for photolithographic mask layout for bulk etching has been extended to allow compensation structures that contain *compound* vertices. The method is general and has been demonstrated to encode existing point solutions and allow for their modifications.

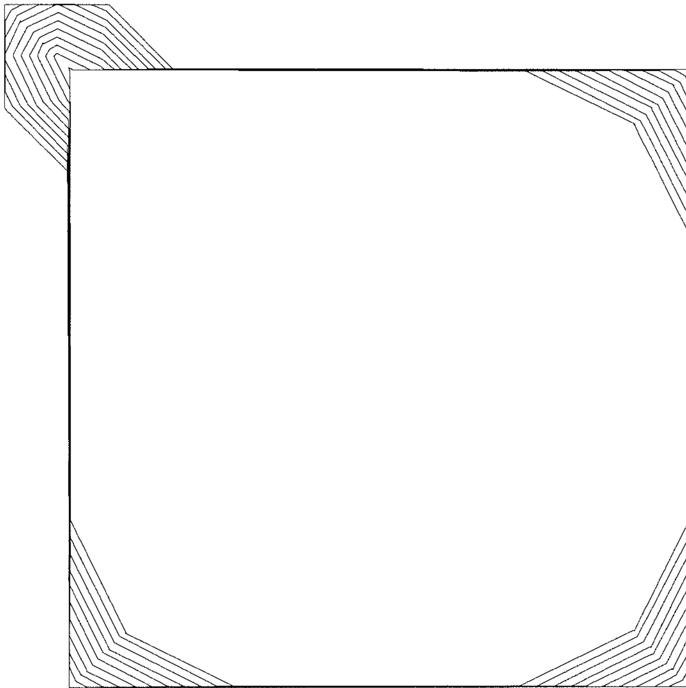


Figure 5.9: Simulation results of computed compensation structure. The other three corners are uncompensated. Outer shape is the mask; inner shape is the result of etching.

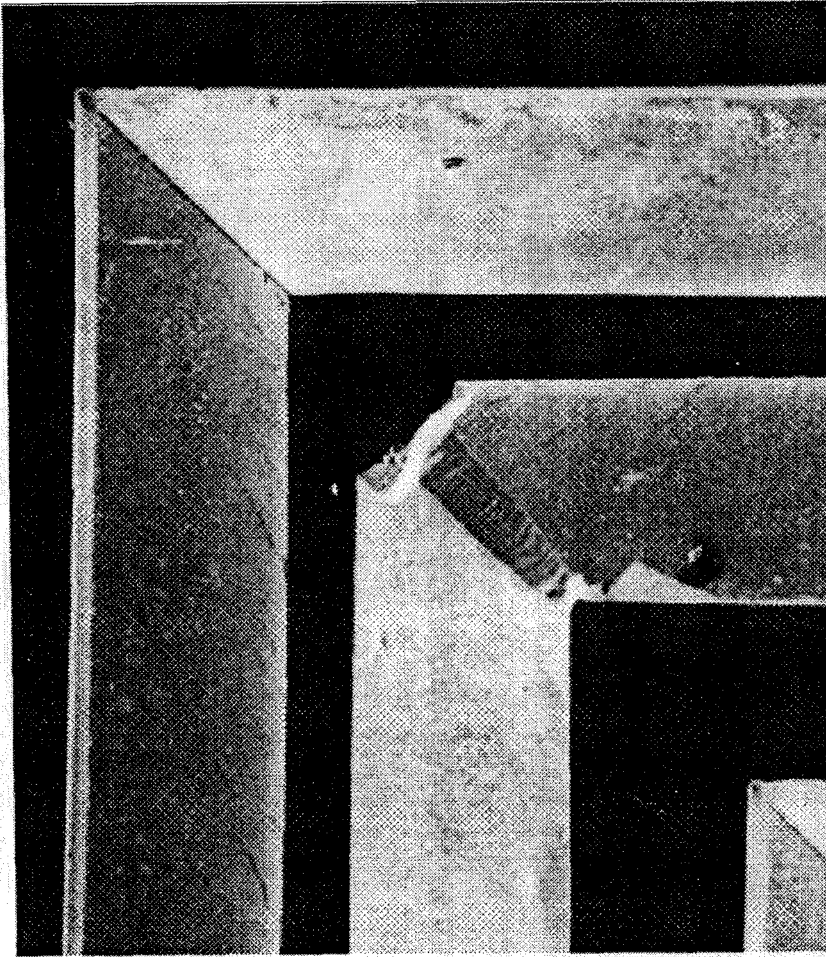


Figure 5.10: Experimental results of compensation structure [65].

Chapter 6

Calibration

One roadblock to increasing the accuracy of bulk etching is the lack of quality process calibration data. The techniques in this thesis require accurate etch rates for the process under consideration. The etch rates have significant variations with respect to temperature and concentration, as well as sensitivities to other process parameters. Previous work in this area is sparse and existing methods relying on voluminous and painstaking quantitative measurements. For example, Zielke et al. [66] rely on 500 or more optical quantitative measurements to determine the rates for orientation dependent etching.

The goal of this calibration technique is to increase the ease and speed of obtaining experimental etch rate data. Also, to consider individual batch calibration as well as "on-line" calibration. To accomplish this goal, accuracy requirements are shifted from the measurement process to the mask design and fabrication process.

The accuracy of any etched geometry is judged by the accuracy of the mask and the accuracy of the measurement. In current practice, the physical photolithographic masks are machine generated, while the etching results are hand measured. To leverage the high accuracy of the machine generated photolithography, the number and accuracy of the masks are maximized to reduce or eliminate the requirement for hand/human measurement.

As the methods of Chapter 4 indicate, for highly anisotropic etchants the local extrema of the etch rate diagram dominate the behavior of the process. Identifying

these values of these extrema etch rates will be the focus of this chapter.

To identify etch rates without quantitative measurements, a large number of test patterns will be employed with slightly varying dimensions. The test patterns are designed such that topological changes indicate specific etch rates and orientations. These topological changes are identifiable by visual inspection and are used to bracket the parameter values. The visual inspection has a binary result (yes/no topology change), and thus has no accuracy requirement.

Slightly different approaches will be required for different wafer cuts. The case of a $\langle 100 \rangle$ wafer (the most common) is considered here.

6.1 $\langle 110 \rangle$ Etch Rate and Orientation

In [15], Ensell employs the test pattern in Figure 6.1 to determine the alignment of the $\langle 110 \rangle$ plane with the wafer. As shown in this figure, the etched holes that are not aligned with the $\langle 110 \rangle$ direction have a slight offset, see also Figure 6.2 center, while the hole and its neighbors that are aligned etch to squares that have no offset from hole to hole; Figure 6.2 right.

While standard wafers include an alignment flat, Figure 2.1, Ensell's verification includes both the accuracy of the flat and the lithography alignment. However, it gives no information about the etch rate.

Whereas silicon $\langle 100 \rangle$ wafers in KOH have very low $\langle 110 \rangle$ etch rates, on the order of $1 \mu m/hr$, other non-silicon mono-crystals with lower anisotropies may benefit more greatly from this test (e.g., Germanium, Figure 3.2). In certain applications, the ratio of etch rates is more important than each specific rate.

Figure 6.3 expands on the pattern of Figure 6.1 with an array of spaced features to determine both the orientation and etch rate for the $\langle 110 \rangle$ plane. By having a large number of circular holes in the mask, each with a different separation spacing, the etch rate can be determined within the interval of the spacing. The accuracy of the method is only limited by the accuracy of the lithography, number of mask holes, and the total etch time. For example, the MUMPS process has a reported

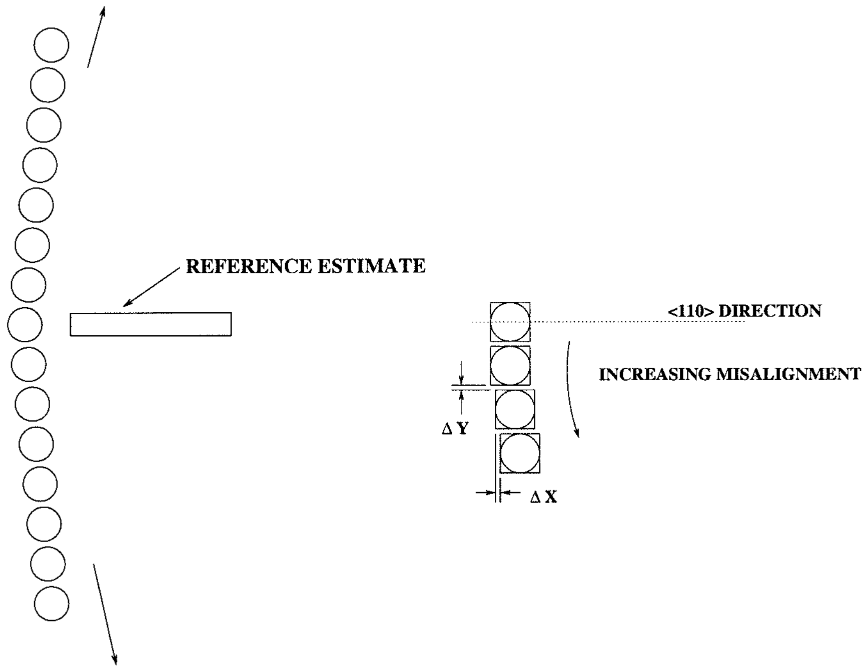


Figure 6.1: (A) Mask layout for ⟨110⟩ orientation determination, (B) variation of results with orientation (after [15]).

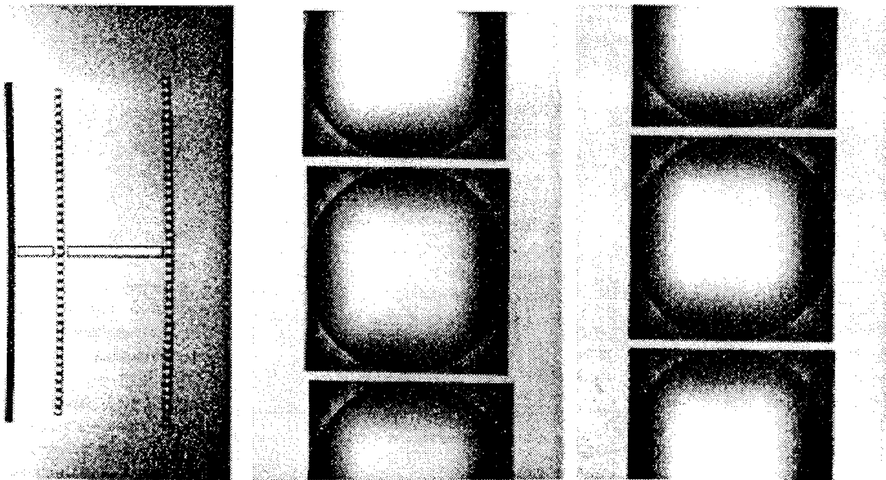


Figure 6.2: Alignment results etched in ⟨100⟩ silicon, from [15].

lithography accuracy of $0.5\mu m$, which if etched for 8 hours would provide an etch rate accuracy limit of $0.0625\frac{\mu m}{hr}$.

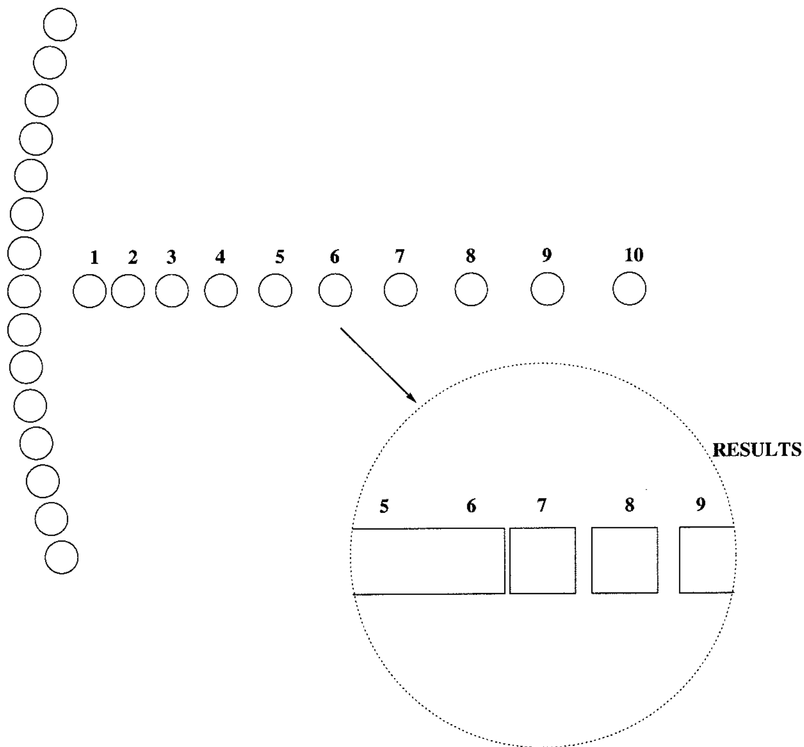


Figure 6.3: Expanded mask pattern for rate determination with schematic of results after etch.

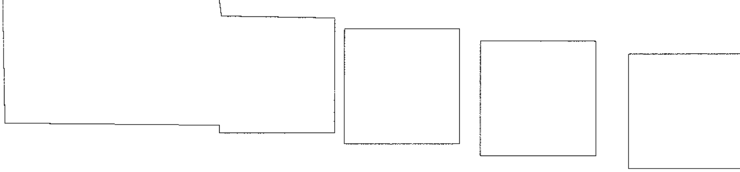


Figure 6.4: Simulation of $\langle 110 \rangle$ rate calibration pattern with 5° misalignment.

The travel of the edge during etching is given by the etch rate multiplied by the etch time. If the combined value for both holes is higher than the separation distance between adjacent holes, the holes will intersect. If the etch rate is lower than the separation, the holes will remain distinct. With a variation in separation distance, the index of the hole at which hole intersection begins indicates the etch rate. In Figure 6.3, visual inspection of the results of the etch reveal that hole intersection begins at hole # 6. The separation distance at index # 5 is $8.5\mu m$ and at # 6 it is $9.0\mu m$. Since our etch time was 8 hours, the etch rate of the $\langle 110 \rangle$ plane ($\langle 111 \rangle$ plane projected onto the $\langle 100 \rangle$ surface plane) is $1.0625\frac{\mu m}{hr} < \text{rate} < 1.125\frac{\mu m}{hr}$.

Since the $\langle 100 \rangle$ Si crystal slice has 4 quadrant symmetry, we can either place 4 times the number of holes, or repeat the pattern 4 times and average the results.

Note also that the results are robust to orientation errors. While the rate calibration pattern is aligned with the *estimated* reference direction, Figure 6.4 shows the SEGS [23] simulation results for a calibration mask whose actual and estimated orientation are misaligned by 5°.

6.2 Fast Plane Etch Rate and Orientation

For etchants such as KOH and EDP, the $\langle 310 \rangle$ plane usually has the highest in-plane etch rate. However, for other etchants, compositions, or non-$\langle 100 \rangle$ wafers, another

Figure 6.5: Simulation of fast plane rate calibration pattern. (A) Mask, (B) Simulation results.

dominant etch plane may need to be determined in the following procedure. Both the etch rate and the orientation of the crystal plane with the globally highest etch rate will be determined.

As was seen in Chapter 4, the crystal plane with the global maximum etch rate is dominant when etching convex surfaces. By masking off circular areas of varying radii, the etch rate can be determined to an accuracy that is given by the discretization of the pattern, Figure 6.5. Again, the circular shape makes this pattern robust to misalignment. In the previous case the rate was determined by hole intersection; here the rate is determined by the elimination of the wafer surface area on top of each circular "peg." In this example, the top circle has a radius of $100\mu m$, for the second circle, $r = 130\mu m$. The etch time was one hour. Since the in-plane surface area of the top circle approaches 0, one can conclude that $100 < \text{rate} < 130$. In practice the gradation between adjacent circle radii would be as small as the lithography would allow. This simulation was done in SEGS 2.5D [23]. The multiple perimeter lines indicate contours of the perimeter sidewalls.

The orientation of the fastest etch plane is also determined by detecting which island or peg has its surface area reduced to zero. In this case, a pattern that is

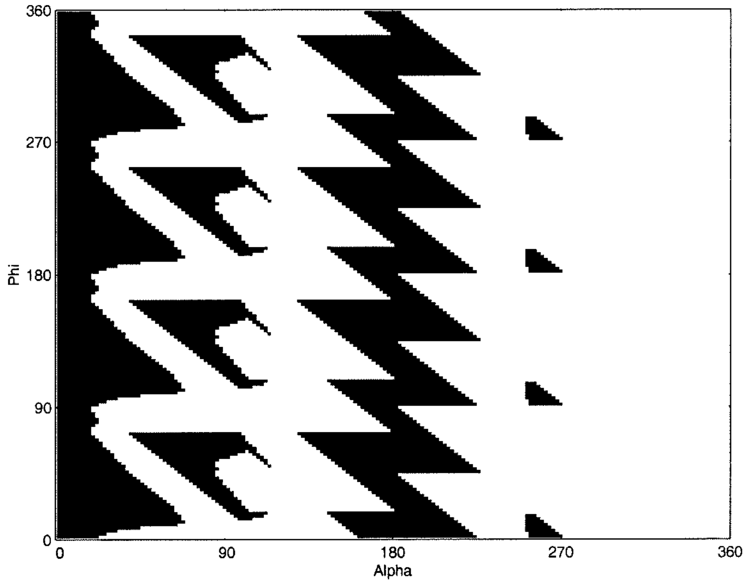


Figure 6.6: Generalized vertex diagram ($\phi - \alpha$).

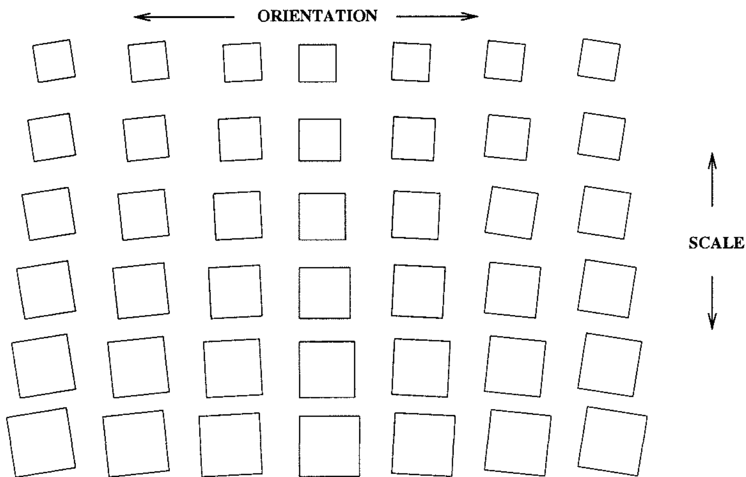


Figure 6.7: Calibration pattern for fast plane orientation determination.

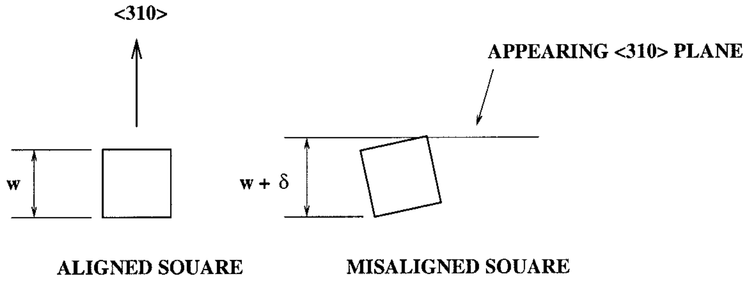


Figure 6.8: Appearance of new planes on misaligned mask.

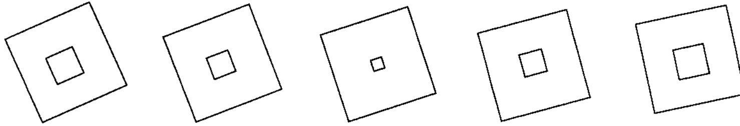


Figure 6.9: Results of Simulation of Calibration Mask.

sensitive to rotational misalignment is used. The change in orientation of a vertex corresponds to shifting its location vertically on the $\phi\alpha$ "generalized vertex diagram" (Figure 4.16), repeated in Figure 6.6. A square mask shape with $\alpha = 90^o$, and $\phi = 82.7^o$ is selected. For clockwise misalignment ($\phi < 82.7^o$), the vertex remains simple, but the vertex velocity is reduced. For counter-clockwise misalignment ($\phi > 82.7^o$), the vertex becomes compound and the appearing plane is the "fast crystal plane" under investigation. However, the plane appears at a distance greater from the center of the square than in the aligned case, Figure 6.8.

The resulting pattern is shown in Figure 6.7. Variations in both size and orientation are used to bracket the estimated values.

Therefore, since clockwise misaligned planes are reduced in speed, and counter-clockwise misaligned planes start further from the center, the aligned square will have its surface area reduced to zero first. A schematic showing the mask shape (outer square) and evolution of the etch are shown in Figure 6.9. Note that the center square is aligned with the $\langle 310 \rangle$ direction and is decreasing in surface area fastest.

6.3 Intermediate Plane

The etch rate diagram for KOH on $\langle 100 \rangle$ wafers, Figure 1.2, has a cusp at 45° due to the rate of the $\langle 010 \rangle$ being slower than the rate in $\langle 310 \rangle$. In order to identify the etch rate, a series of diamond shaped holes with varying widths are placed as shown in Figure 6.10. The etch rate is determined by identifying which hole in the array has evolved to a square, e.g., $0.7071 * w_i < r < 0.7071 * w_{i+1}$, where r is the etch rate of the $\langle 010 \rangle$ plane and w_i is the width of the ith diamond in the mask layout.

6.4 Summary

This chapter presented an approach to determining the etch rates and orientations for the three dominant crystal planes in anisotropic etching with KOH-like etchants. The key idea is to shift the accuracy requirements from the measurement process to the mask design and fabrication process. There are many possible mask patterns that could be used for calibration; the ones presented in this chapter were meant to be illustrative, not necessarily optimal. The etch rates of planes not aligned with these principal directions are determined by interpolation. Additional patterns may also be used to refine the interpolation functions used by directly identifying the rates of intermediate planes.

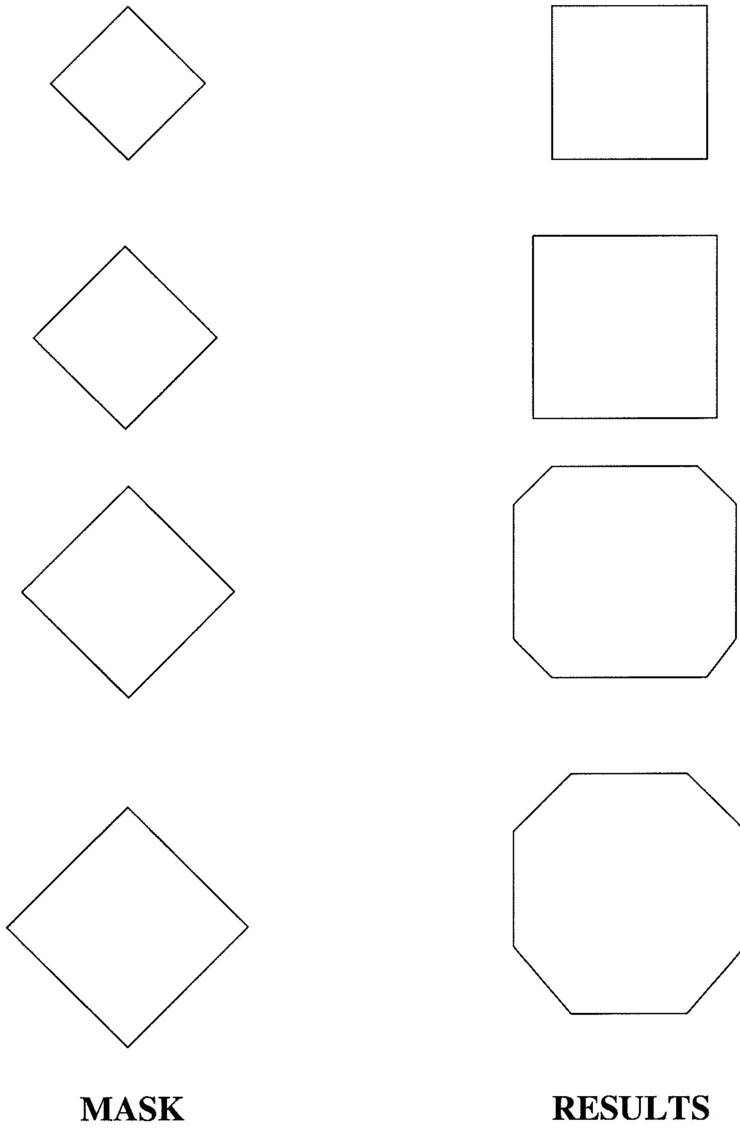


Figure 6.10: Mask and simulation results for $\langle 010 \rangle$ rate calibration.

Chapter 7

Algorithm Design and Implementation

In this chapter the implementation of the algorithms presented in Chapters 4 and 5 will be discussed. The synthesis algorithms of Chapter 4 are based on modifying local feature vertex topology iteratively until a set of new simple vertices is reached, thereby allowing direct preimages (reverse simulations) to be calculated. For a desired total etch time, the Timing Models in Chapter 5 are solved to yield the coordinates of the preimage vertices. No reverse simulation is required.

Following the discussion of the core algorithms, an overall implementation strategy is described that illustrates how these algorithms may be used in an integrated computer-aided micro machining package. The system is divided into two pieces, the "computational engine" implements the algorithms discussed so far (the server), and the user interface that handles user input and interaction, and reporting and displaying of results (the client).

The algorithms presented were designed based on an object oriented model and implemented in the Java programming language. Java was chosen for several reasons: graphics routines native to the language made debugging computational geometric code easier, graphical user interface (GUI) primitives native to the language made interactivity available at an early stage in the development, code portability allowed for the future possibility of deployment on the internet, and the implicit garbage collection native to the language improved debugging time by eliminating pointer bugs common in C and C++ programs.

The design of a two part client-server system fit naturally with the choice of the Java language. By only having the user download the interface, the download time is reduced and the computational requirements of the users systems are reduced. Additionally, since all the computational code (the server) remained at the host site, it can be continually maintained and improved. This also provides some level of enhanced intellectual property protection.

Instead of stepping through the software system sequentially, it will be presented in a hierarchical manner starting with the polygon that represents the desired resultant geometry of the silicon after the specified etching process.

7.1 Polygon Layer

The polygon geometry that is the desired result of the etch is first decomposed into its constituent vertices. A preimage is developed for each vertex individually, Figure 7.1. The candidate solution for each feature vertex is represented by either vertex velocity vector(s) or the coordinates of the preimage vertices (in the case of a timing model solution). Equation 3.11 is used to calculate T^*, the time at which a line segment in the reverse propagation of the topology vanishes. This is compared to T_d, the desired etch time.

If $T^* >= T_d$, for all preimage vertices, there are no interactions between the preimages of adjacent vertices. The desired etch time may be applied to the vertex velocity vectors to calculate the coordinates for each vertex in the mask layout preimage.

If $T^* < T_d$, there are feature interactions between the individual preimages. Hence, the loop is repeated, Figure 7.1, with the methods of § 4.1.3 used to find an alternative preimage which reduces g'_{i+1}, Equation 3.3. If the interactions can not be eliminated, the designer will make the final decision on the appropriate geometry modifications.

Once the interactions are minimized, the preimage for the polygon is complete. Note that it is possible that adjacent vertex interactions cannot be eliminated, either

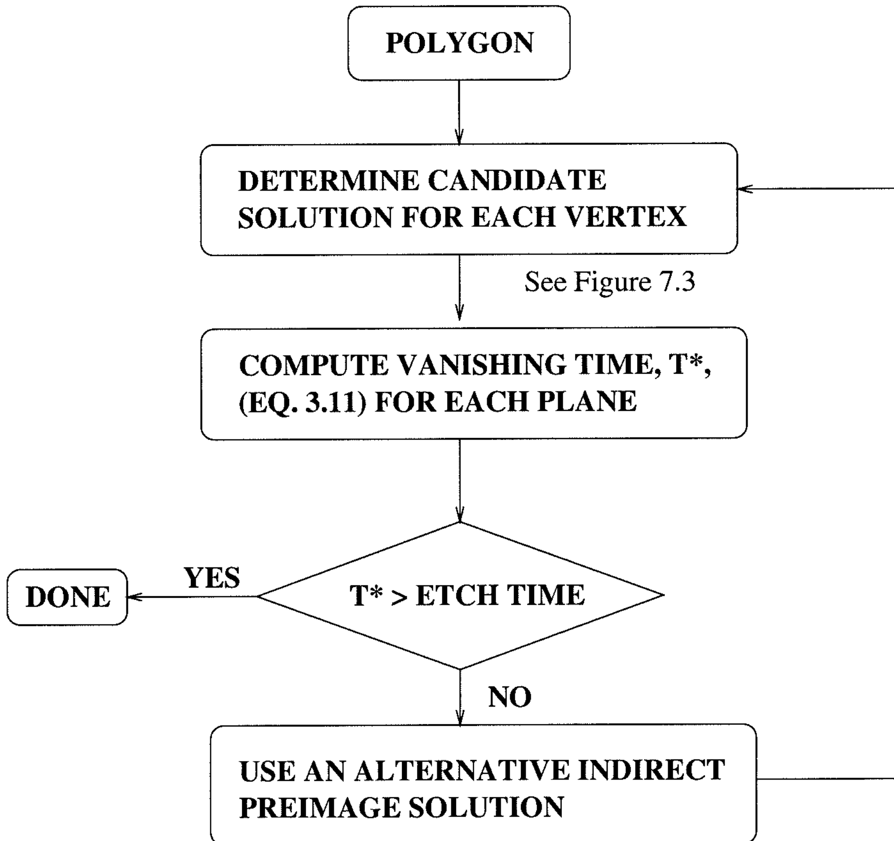


Figure 7.1: Algorithm for decomposing vertices of each polygon.

the precise desired geometry cannot be fabricated or more sophisticated masks may be required. In either case the designer will be provided the details of the interaction and called upon to resolve the conflict.

7.2 Vertex Classification

The first step in the calculation of a preimage for a feature vertex is vertex classification. That is, each vertex must be determined to be concave or convex and either *simple* or *compound*, Definitions 3.7 and 3.8.

The vertex is determined to concave or convex simply by calculating the magnitude of the spanned angle. While there are two sides to every vertex, the convention used throughout this thesis has been that the side of the vertex exposed to the

etchant is used for this calculation, Figures 4.1 and 4.9.

The vertex's *simple/compound* classification is based on whether new plane(s) would appear at this vertex if it were etched with its current span and orientation. See Definitions 3.7 and 3.8.

As discussed in § 3.2, the *simple/compound* classification is computed using the convex hull of the portion of the reluctance diagram bound by the reluctance vectors of the planes that form the vertex, Figure 7.2. The convex hull calculation is the most computationally intensive segment of the algorithm. The convex hull algorithm used in this thesis was from Jeff So [53]; it was selected primarily due to the availability of the Java source code.

7.3 Preimage Calculation Algorithm

The methods developed in this thesis provide up to three options to compute a preimage for a feature vertex, depending on its classification.

7.3.1 Indirect Preimage Algorithm

The indirect preimage solution provides a structured approach to computing multiple inverse vertex velocity vectors for a single feature vertex. This results in a single feature vertex having multiple corresponding vertices in the mask layout, Figure 7.3. Under a timed etch, all the forward vertex velocity vectors meet at the single desired vertex in the desired resultant geometry, Figure 3.10. Since multiple vectors meet precisely at the end of the etch, the line segment bounded by two adjacent vectors vanishes exactly at the end of the etch.

For compound concave vertices, the parameterized compensation structures of § 4.3 are used to determine the set of vertex inverse velocity vectors. For simple vertices, indirect preimages are used to include additional designer requirements, § 4.1.3.

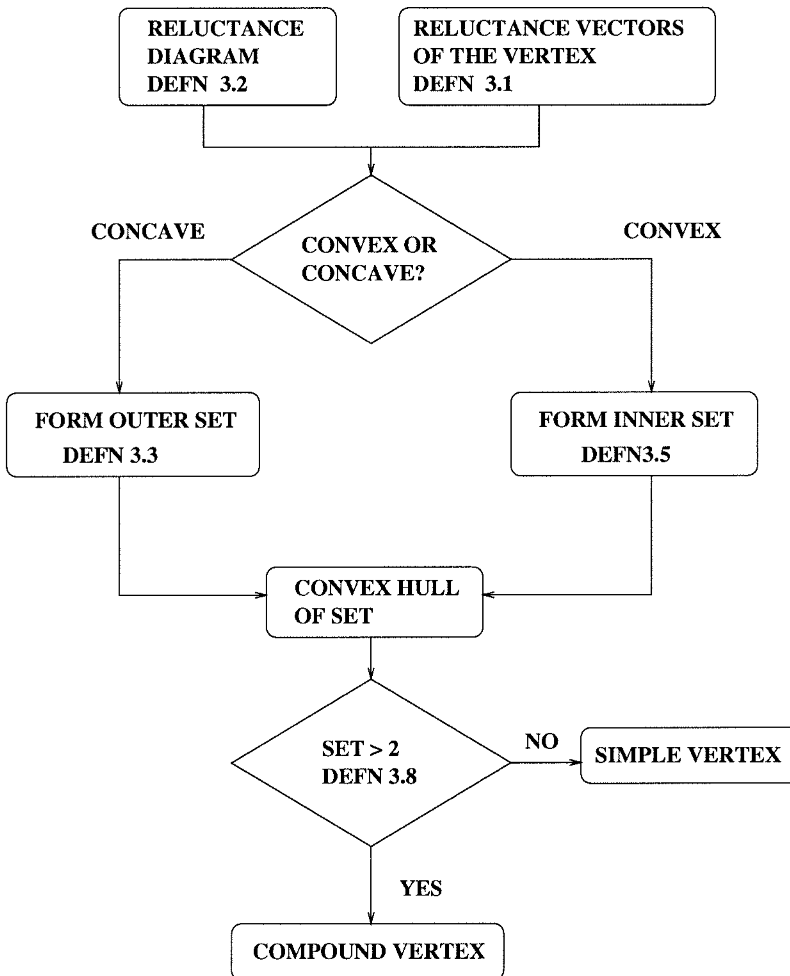


Figure 7.2: Algorithm for convex hull calculation.

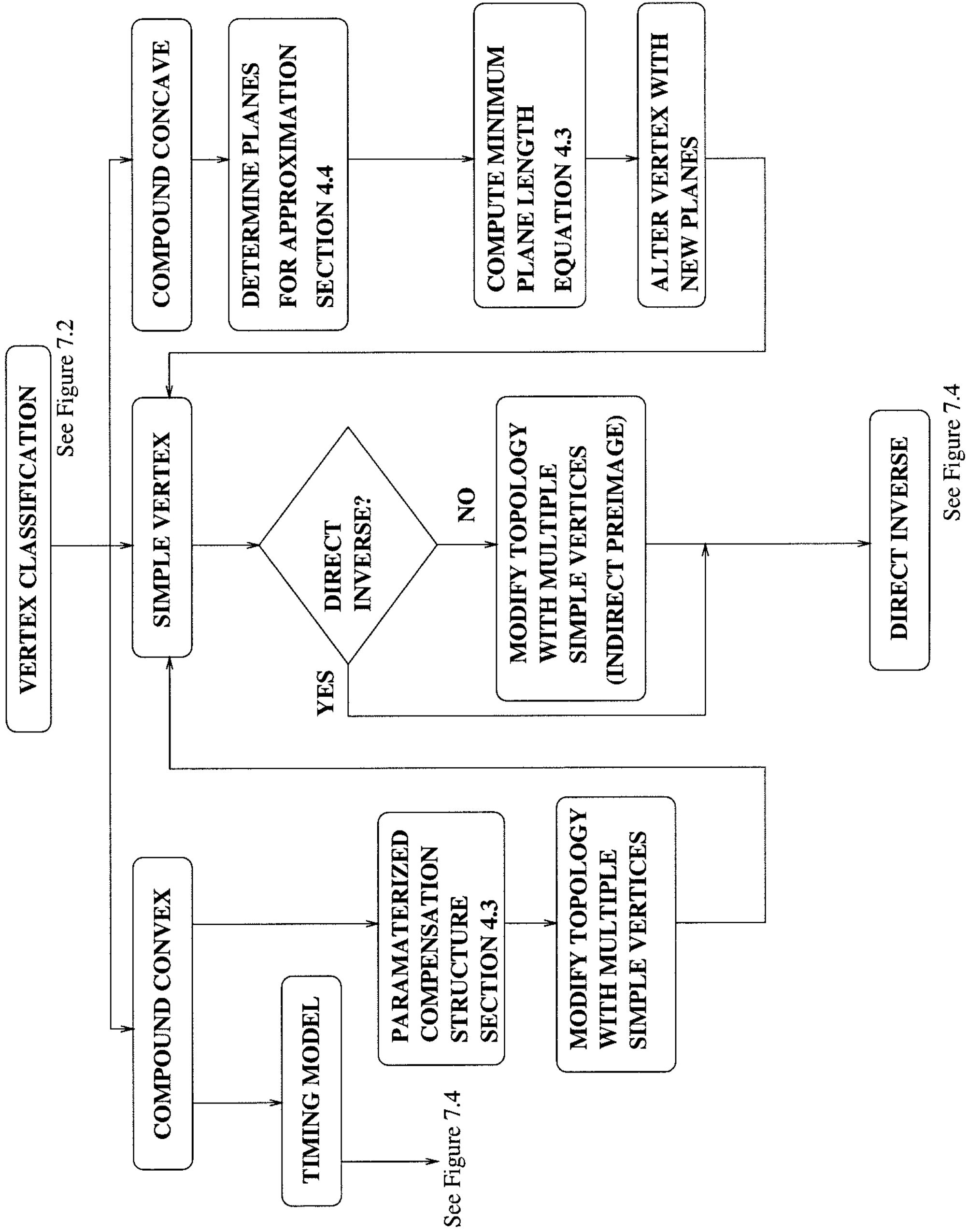


Figure 7.3: Algorithm for the vertex classes.

7.3.2 Direct Preimage Algorithm

For feature vertices of type simple, the direct preimage is computed using a reverse simulation approach discussed in § 3.2 and Equation 3.13; See Figure 7.4. This calculation requires the knowledge of the etch rates in the directions normal to the two planes that form this vertex. These etch rates encompass the properties of the material being etched, the composition and concentration of the etching solution, and the process parameters (temperature, stirring rate, batch size, etc). While a number of approaches were discussed in Chapter 2, the approach that was implemented was based on linear interpolation in (r, θ) coordinates between the $\langle 310 \rangle$, $\langle 110 \rangle$ and $\langle 100 \rangle$ basis etch rate vectors [23].

7.3.3 Timing Model

The timing model method developed in Chapter 5 is based on encoding specific preimages in a system of equations. Once the preimage topology is encoded, the etch time, and etch rate parameters are used to solve for a specific preimage. Therefore, by definition, when employing a timing model preimage for a specific feature vertex, the designer will select from a menu of predefined preimage topologies, Figure 7.4. The coordinates of the mask layout vertices that constitute the preimage for that individual feature vertex will result.

7.4 Design and Analysis Plotting Algorithms

Throughout the thesis several data plots have been presented to illuminate properties of a particular etchant or to aid in design tradeoffs during preimage calculation. This section discusses the algorithms used in the computation of each plot.

7.4.1 $\phi\alpha$ Diagram

The $\phi\alpha$ plot, first shown in Figure 4.16, is simply a plot of the classification of each possible vertex. A generic vertex is parameterized by its orientation with respect to the crystal, ϕ, and angle of inclusion, α, Figure 4.15. By incrementally sweeping

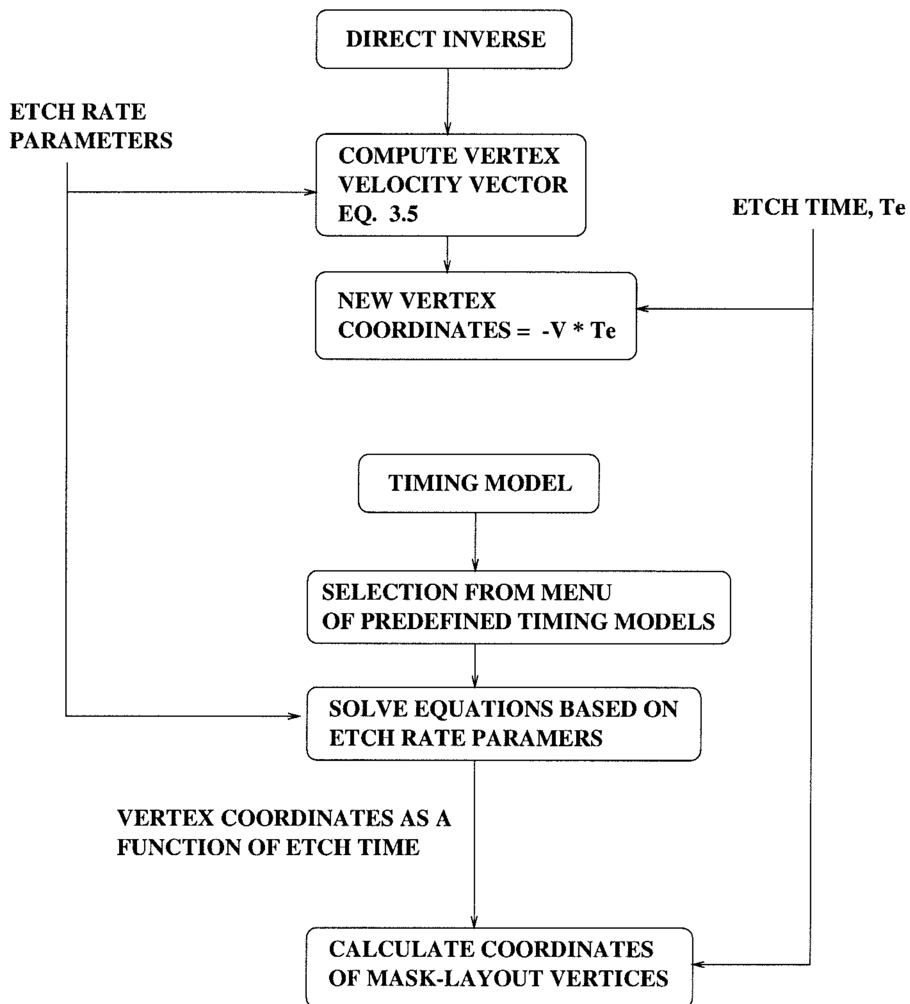


Figure 7.4: Algorithm for direct inverse and timing model final calculation.

ϕ and α through their full range of values, $[0, 2\pi)$, and classifying the vertex as *simple* or *compound* at each point, Figure 7.2, a plot of the space of *simple* and *compound* vertices is generated. When a (ϕ, α) coordinate corresponds to a vertex that is *simple*, that point on the plot is colored black, when the corresponding vertex is *compound*, the point is colored white. This plot relies on the complete two-dimensional etch rate diagram for the process under consideration. Changes in process parameters will directly alter the plot results.

In § 8.1.1, Figure 8.2, this plot is augmented by calculating the tangential component of the vertex velocity for compound vertices. This in addition to the regions of *simple/compound* differentiation, there is additional data about the behavior of the appearing plane.

7.4.2 $\gamma-$ Velocity Plot

In § 4.1.3, the $\gamma-$Velocity plot was used to choose an indirect preimage solution to satisfy an additional constraint. The plot in Figure 4.7 was generated by considering the orientation, γ, of all possible planes that could be inserted at the feature vertex. The variable γ was swept through its range and the tangential velocity of the vertex velocity vector was plotted on the abscissa. Note that only certain values of γ correspond to valid indirect preimages, so only the velocity at those values was plotted.

7.4.3 $\beta_2\beta_3$ Diagram

In § 4.3, Figure 4.17, $\beta_2\beta_3$ plots were used to present the space of all valid three vertex compensation structures. The candidate compensation structure was fully parameterized by two variables β_2 and β_3, Figure 4.15. Both variables were swept through the range $[0, \pi)$, at each point the three vertices of the candidate compensation structures were validated to be of class *simple* and the growth rate, g' Equation 3.8, of line segments of the compensation structure, $\overline{A'_1A'_2}$ and $\overline{A'_1A'_3}$, Figure 4.15, were verified to be positive. $g' > 0$ for each plane or line segment of the compensation structure indicates that it will grow in size and not collapse at

$t = 0^{-}$. These conditions indicate that the three simple vertices that replace the one compound convex vertex will form a valid preimage for the feature vertex. The coordinate of the plot is given a grey color if these conditions are met and thus the values of β_2 and β_3 correspond to a valid preimage.

Additional information can now be added to the plot to differentiate the various valid preimages. In Figure 4.17, the normalized distance from the desired vertex, A, to the point of the preimage, A_1', was used to determine the grey-scale value used, Figure 4.15.

7.5 User Interface and High Level Structure

For reasons of completeness, this section will discuss the design of the user interface to the algorithms discussed in the preceding sections of this chapter. The software system is called "CAMLE," Computer Aided Mask-layout for Etching (pronounced "camel"). Similar to the Computer-Aided Machining (CAM) packages of today (MasterCAM, etc.), it is likely that there will soon be commercially available Computer-Aided Micro-Machining software packages (μCAM or micro-CAM), of which the technology developed in this thesis and embodied in CAMLE will be useful.

As discussed in the introduction to this chapter, the algorithms were designed based on an object oriented model and implemented in the Java programming language. Object specifications are presented in Appendix ?? for those interested in that level of detail. Java was chosen for several reasons, graphics routines native to the language made debugging computational geometric code easier, graphical user interface (GUI) primitives native to the language made interactivity available at an early stage in the development, code portability allowed for the future possibility of deployment on the internet, and the implicit garbage collection native to the language improved debugging time by eliminating pointer bugs common in C and C++ languages.

7.5.1 Client-Server Architecture

Early versions of CAMLE were developed using Sun Microsystems Appletviewer for execution, due to rapidly expanding complexity of the software system and time delay between the release of Java Virtual Machines from Sun and their inclusion in Netscape browsers, execution in a browser soon became unbearably slow. As discussed in the introduction to this chapter, moving to a two part client-server architecture fit naturally with the desire to deploy the system on the internet. By only having the user download the interface, the download time is reduced and the computational requirements of the users systems are reduced. Additionally, since all the computational code (the server) remained at the host site, it can be continually maintained and improved.

The graphical user interface (GUI) and the computational engine would need to communicate over the internet using a standard protocol. Several were available to choose from. At the time of CAMLE's development, Java's support of CORBA (Common Object Request Broker Architecture) was not complete. Since CAMLE's needs were not very sophisticated, JAVA's Remote Method Invocation (RMI) was chosen as an interface protocol. RMI allowed the computation algorithms to be placed on a server waiting to service requests for computation from the internet. On the client side, the GUI could call computation routines exactly as if there were linked into the local code base. The Java RMI library would locate the address of the CAMLE RMI server and seamlessly transport the function parameters from the GUI to the server and return the results. Using RMI, all computation was placed on the server and the amount of code the user was required to load over the internet and execute in the local browser was minimized, Figure 7.5.

During the early stages of development, the SEGS bulk etching simulator [23] was used to validate some of the algorithms and preimage calculations. To make this testing easier, the GUI could request to send simulation data to the SEGS simulator instead of the CAMLE simulator (while CAMLE's main task is mask synthesis, the core CAMLE algorithms allowed an unsophisticated simulator to be implemented). Upon receipt of a request for a SEGS simulation, the computational server would

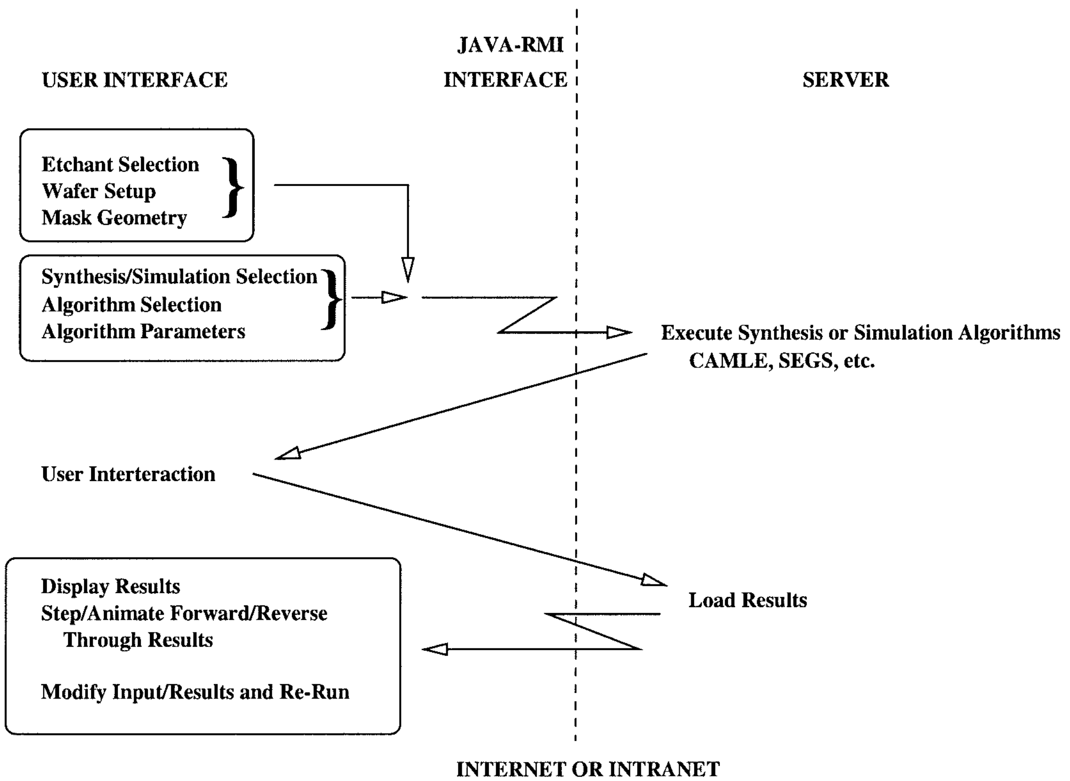


Figure 7.5: CAMLE Operation.

then generate the necessary data files for SEGS and execute a SEGS process in batch on the server computer.

7.5.2 User Interface and Operation

To compute a mask layout, CAMLE requires etch rate data and a description of the polygon that is the desired result of the etch. To enter etch rate data, the user could select from list of common etchants or enter a URL to any file on the internet that contained etch rate data in the specified format. Once loaded, the display showed the user a two-dimensional etch rate diagram for visual inspection. Similarly, the desired shape to be etched could be selected from a local library, or any valid URL could be specified. Note that for users to specify data files on the internet, these files must be in publicly accessible directories on the internet. Both secure downloads and FTP access were considered, but not deemed worthy of effort in early versions of the systems.

Once etch rate and geometric data were entered into CAMLE, the user could select to simulate etching (using SEGS or CAMLE algorithms) or synthesize a mask layout using the algorithms developed in this thesis. After specifying the desired etch time and display time step and clicking on the execute button, the data is passed to the computational sever using RMI as discussed above and the results were returned. The user could now step through the forward etch or the "backward" mask synthesis one frame at a time, or animate continuously at a specified rate.

7.6 Summary

This chapter discussed the design and implementation of the mask layout synthesis algorithms developed in this thesis.

Chapter 8

Conclusions and Future Work

While there has been a variety of work in the area of computer aided design for micro-electrical-mechanical systems (MEMS) and significant work in the area of simulating the behavior anisotropic bulk etching, this thesis has presented the first systematic research aimed at directly computing the photolithographic mask layout geometry for bulk etching.

A classification system was devised that described the behavior of planes and vertices during the etching process. This classification was then used to describe methods to compute preimages for each vertex that would etch to the desired shape. It was shown that while vertices of type *simple* have a trivial *direct preimage*, they also have a multitude of alternative solutions computable using the *indirect preimage* approach. Further, it was shown how *compound convex* vertices could be decomposed into multiple vertices of class *simple.* When the preimage for this new set of vertices was computed, the resulting mask layout would etch the desired compound convex vertex. The resulting mask shape formed by multiple direct preimages of simple vertices is called a compensation structure. Vertices of type *compound concave*, with no exact preimage, were solved using an approximation method that used the inevitably appearing planes to modify the local topology. The new topology was composed of multiple simple vertices that could then be solved using the other methods. All of these methods replaced each simple vertex with multiple vertices that were ultimately solved using the direct preimage approach.

To facilitate the computation of preimages with compound vertices, the timing model approach was developed. A compound vertex has appearing planes; therefore, any synthesis calculation must have the appearing planes disappear exactly at the end of the backward propagation. The structure of the preimage is encoded in a system of equations that represent the appearing and disappearing of the various crystal planes during the etch. Once the structure is defined, the etch process properties are included and the system is finally solved for the desired etch time.

Since all of the mask synthesis development relies on accurate etch process data, techniques for deriving etch rate data were examined. Currently process calibration approaches rely on painstaking quantitative measurements. A method was developed that leverages the highly accurate machine generated photolithographic etch masks to eliminate the need for quantitative measurements. A mask pattern that is composed of a large number of slightly varying geometries is produced, the etch results are then visually inspected and the location of specific topological variations within the pattern are used to identify key etch parameters.

8.1 Areas for Future Development

This section discusses some avenues for potentially fruitful further research to extend the work developed in this thesis.

8.1.1 Intelligent Geometry Modification

During the development of part geometry and the synthesis of the mask layout, there are occasions in which the designer may desire to modify the shape that is to be etched. There is the potential to develop tools to guide this modification process. Figure 8.1 shows the $\phi\alpha$ diagram developed in § 4.3. This plot may be augmented with a variety of helpful parameterizations. Figure 8.2 shows the $\phi\alpha$ diagram where the colorbar gives the tangential component of the vertex velocity of the appearing plane. Figure 8.3 shows the points corresponding to the vertices of a particular geometry. For clarity, the underlying $\phi\alpha$ data from Figure 8.1 is removed. Note

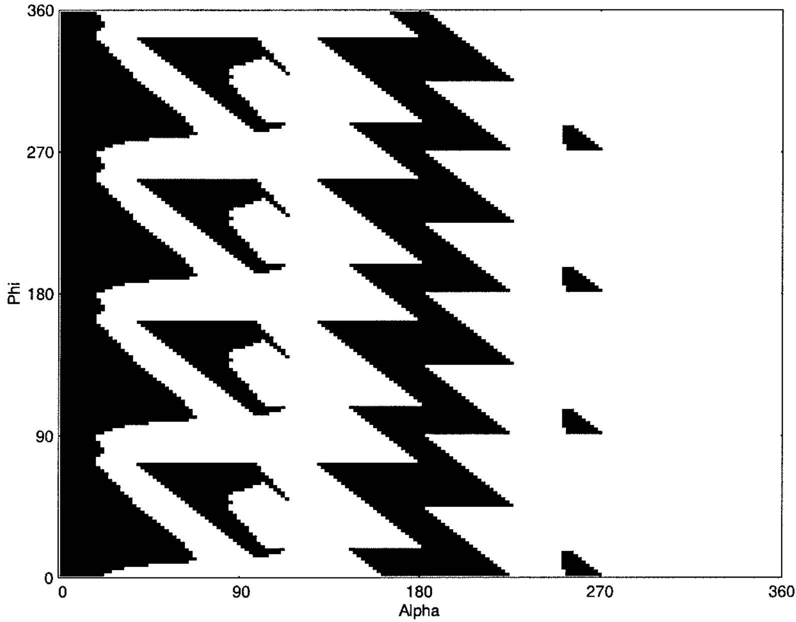


Figure 8.1: $\phi\alpha$ Diagram Shows Classification of a Generalized Vertex.

that scaling or translating the geometry results in no change in the geometry of the $\phi\alpha$ diagram. Displacement of the diagram along the ϕ axis (vertically) corresponds to rotating the shape with respect to the crystal. Movement of one data point along a diagonal of the diagram corresponds to changing the topology of one vertex while leaving the next vertex unchanged (ϕ_{i+1}, α_{i+1} = constant).

This is one example of using the $\phi\alpha$ diagram to alter the part geometry in a directed fashion. There is the potential to develop structured or guided methods to alter the resultant shape to improve manufacturability with a particular etchant and process.

8.1.2 Improved Calibration

While the basic idea for the calibration technique has been verified by simulation, there is still room to improve the calibration mask patterns so as to provide more accurate and easier identification of etch parameters. Additionally, only the rate and direction of the three principle rates were identified in the proposed scheme. Pat-

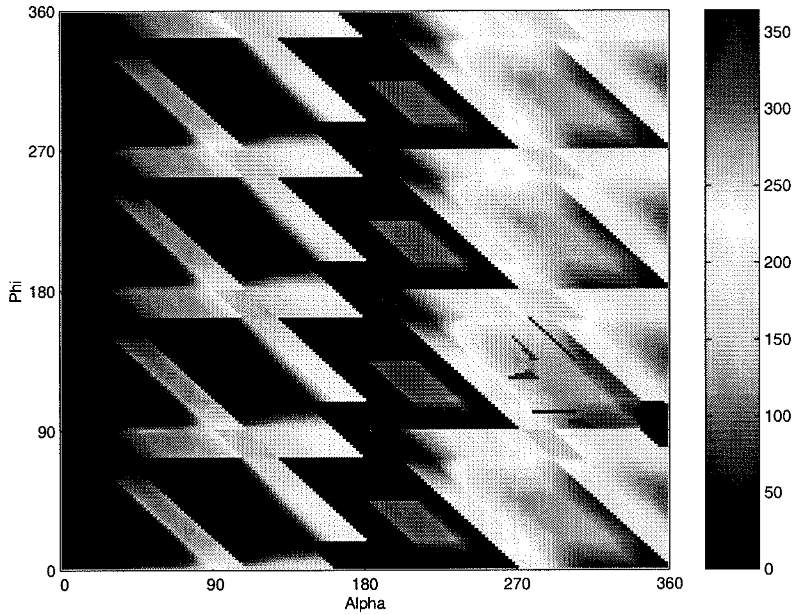


Figure 8.2: $\phi - \alpha$ Diagram Augmented with Appearing Plane Rate Information.

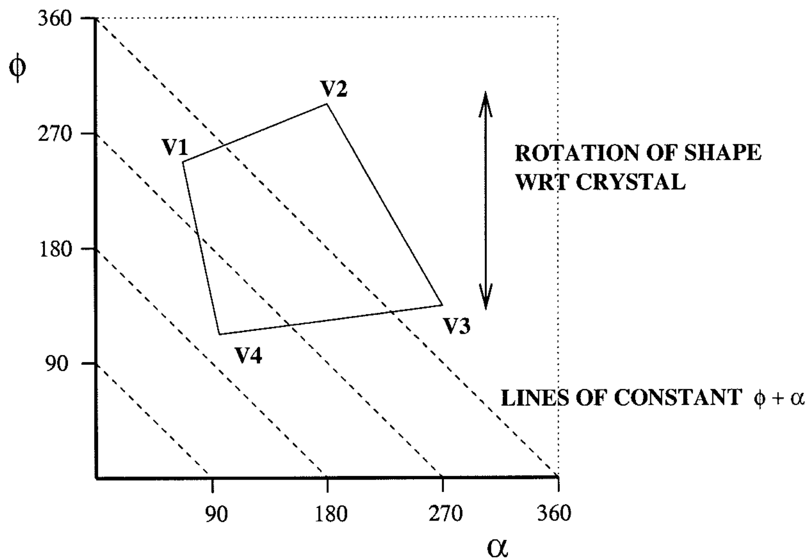


Figure 8.3: Part Geometry Overlayed on $\phi - \alpha$ Diagram.

terns to identify intermediate rates that could be used to validate the interpolation functions used to generate the etch rate diagrams would also be useful.

8.1.3 Multi-Polygon Mask Layouts

The work thus far has developed masks layouts in which each polygon in the desired output has a single corresponding polygon in the mask layout. Recent work by Nikpour, Landsberger, and Hubbard [36], Figure 8.4, has demonstrated improved approximation for *compound concave* vertices using a mask layout composed of many small "cells." When etched, these small polygonal "cells" interact to delay the formation of the unwanted planes in the result.

Extending the timing model development from Chapter 5 to include the formation of multiple "cells" could broaden the space of available mask layout solutions. The basic crystal pattern used in this mask layout, Figure 8.4, is the one developed in Chapter 5, Figure 5.3. Each interaction of two cells creates a new instance of the geometry in Figure 5.3.

8.1.4 Three-Dimensional Micro-Structures

The work presented thus far has described the evolution of two-dimensional patterns of etch shapes on the surface of a mono-crystalline wafer. As the patterns are etched along the surface they are also etched *into* the surface, Figure 8.5. While some have studied simulation methods for general three-dimensional etching [51, 55], the focus here has been the shapes that result from photolithographic masking. That is, those three-dimensional contours that start as a two-dimensional pattern on a flat surface of a mono-crystalline wafer. While it is a subset of the general three- dimensional problem, it is consistent with application domain under consideration in this thesis.

This restriction provides an important simplification of the general three- dimensional problem. As discussed in the previous chapters, appearing planes are planes with locally maxima or minima etch rates (dominant crystal planes). However, the orientation of surface edges is determined by the photolithography and has no in-plane restrictions. The result is that the sidewall angle is determined solely by the

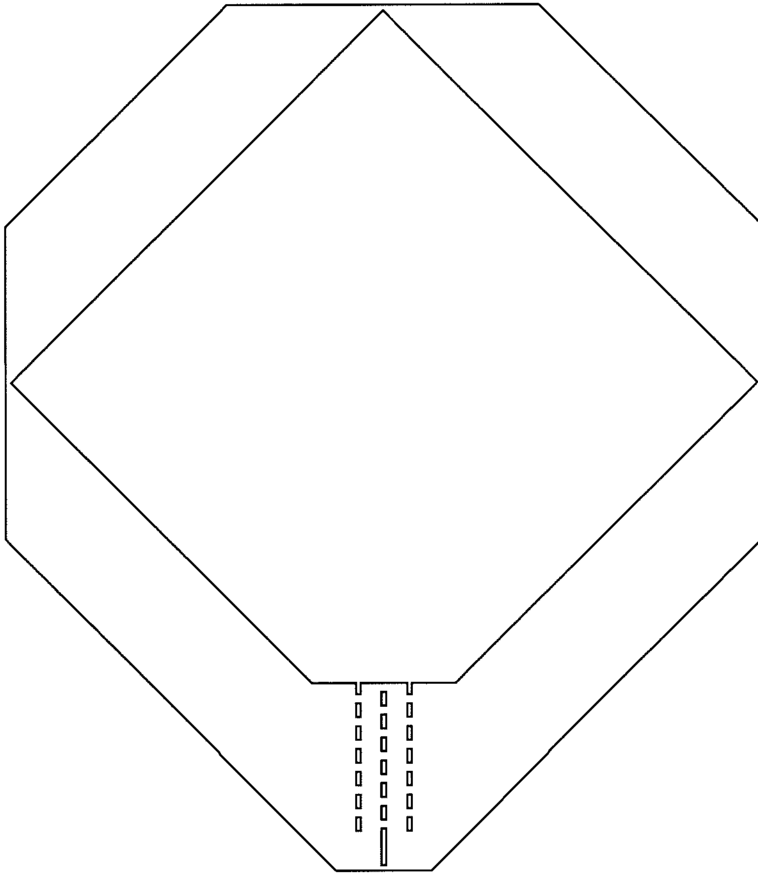


Figure 8.4: Mask layout for improved compound concave vertex [36], outer polygon is etch result.

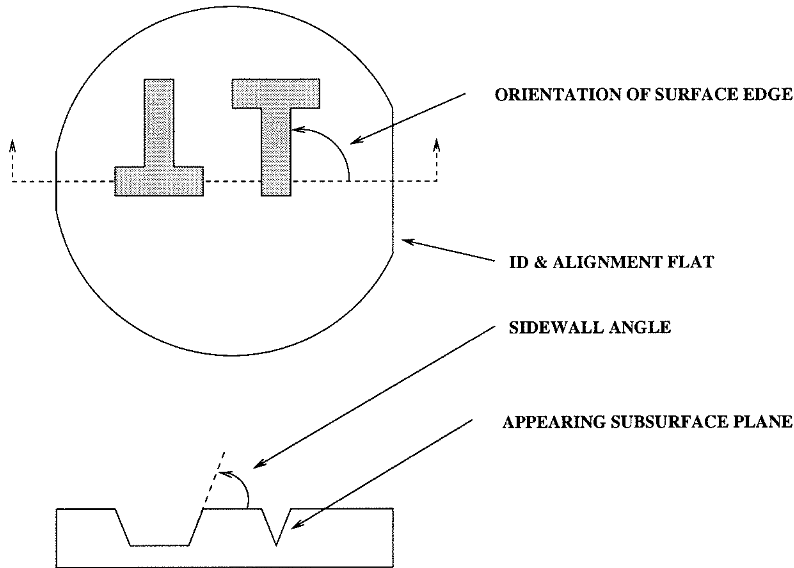


Figure 8.5: Surface Edges and Appearing Sidewalls.

etch rate diagram and the orientation of the surface edge, Figure 8.5. Additionally, any plane that may appear during the etching will be a locally dominant crystal plane.

Figure 8.6 indicates the sidewall angle vs. orientation for a ⟨100⟩ Si wafer with the etch rate diagram of Figure 1.2. This data was extracted from the results of simulations using SEGS [24, 23].

Due to the structure of the crystal and properties of the wet etching process, the surface orientation of an edge and the vertical sidewall angle cannot be independently selected. This lack of independence must be taken into account when synthesizing the desired surface pattern. Once the surface pattern is selected, the methods of Chapter 4 can then be used.

The two-dimensional etch models (§ 3.1) that were the basis for the previous development have direct extensions to three-dimensions [51]. The vertices are now formed by the intersection of three or more planes, and appearing planes are determined by the corresponding three dimensional convex hull of reluctance vectors. Since the only way for a non-surface edge to form is by an appearing plane, the in-wafer contour of an etch is restricted to dominant crystal planes (appearing planes).

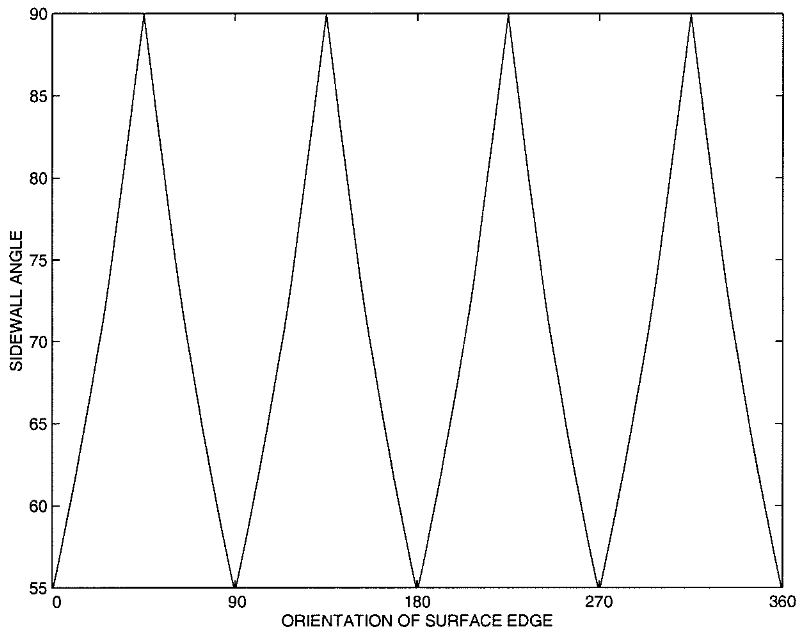


Figure 8.6: Appearing Sidewall Angle vs. Surface Orientation.

The vertex taxonomy laid out in § 3.2 can now be expanded to accommodate three-dimensional features.

Bibliography

[1] M.M. Abu-Zeid. Corner undercutting in anisotropically etched isolation contours. *J. Electrochem. Soc.*, 131:2138–2142, 1984.

[2] D. Adalsteinsson and J.A. Sethian. A level set approach to a unified model for etching, deposition, and lithography i: Algorithms and two-dimensional simulations. *Journal of Computational Physics*, 120:128–144, 1995.

[3] H. Baltes. Cmos micro electro mechanical systems. *Sensors and Materials*, 9(6), 1997.

[4] M. Bao, C. Burrer, J. Esteve, J. Bausells, and S. Marco. Etching front control of (110) strips for corner compensation. *Sensors and Actuators*, A37-38:727–732, 1993.

[5] B. W. Batterman. *J. Appl. Phys.*, 28(1236), 1957.

[6] I. Brodie and J.J. Muray. *The Physics of Microfabrication.* Plenum Press, New York, 1982.

[7] R. A. Buser, S. B. Crary, and O. S. Juma. Integration of the anisotropic-silicon-etching program ASEP within the CAEMEMS CAD/CAE framework. In *MEMS 92*, pages 133–138, Institute of Electrical Engineers, 1992.

[8] R. A. Buser and N. F. de Rooij. ASEP: A CAD program for silicon anisotropic etching. *Sensors and Actuators*, 28:71–78, 1991.

[9] Rudolf A. Buser and Nicolaas F. de Rooij. Asep: a cad program for silicon anisotropic etching. *Sensors and Actuators A*, 28:71–78, 1991.

[10] H. Camon and A. Gue. Modelling of an anisotropic etching in silicon-based sensor applications. *Sensors and Actuators*, 33:103–105, 1992.

[11] R.A. Colclaser. *Microelectronics: Processing and Device Design.* John Wiley & Sons, New York, 1980.

[12] S. B. Crary and Y. Zhang. CAEMEMS: An intergrated computer-aided engineering workbench for micro-electro mechanical systems. In *MEMS 90*, pages 113–115, Institute of Electrical Engineers, 1990.

[13] J.S. Danel and G. DeLapierre. Anisotropic crystal etching: A simulation program. *Sensors and Actuators*, 31:267–274, 1992.

[14] M. Elwenspoek. On the mechanism of anisotropic etching of silicon. *J. Electrochem. Soc.*, 140:2075–2080, 1993.

[15] G. Ensell. Alignment of mask patterns to crystal orientation. *Sensors and Actuators A*, 53:345–348, 1996.

[16] Bill Foote. Simulation of anisotropic crystal etching. Master's thesis, U.C. Berkeley, 1990.

[17] F. C. Frank. On the kinematic theory of crystal growth and dissolution processes. In *Growth and Perfection of Crystals; Proceedings of an International Conference on Crystal Growth*, pages 411–419. John Wiley and Sons, 1958.

[18] F. C. Frank and M. B. Ives. Orientation-dependent dissolution of Germanium. *J. Appl. Phys.*, 31(11):1996–1999, November 1960.

[19] J. Fruhauf, K. Trautman, J. Wittig, and D. Zeilke. A simulation tool for orientation dependent etching. *J. of Micromechanics and Microengineering*, 3:113–115, 1993.

[20] S.K. Ghandi. *VLSI Fabriation Techniques.* John Wiley & Sons, New York, 1983.

[21] C.E. Hallas. Electropolishing silicon. *Solid State Tech.*, 14:30–32, 1971.

[22] Mohammed Hasanuzzaman and Carlos H. Mastrangelo. Process compilation of thin film microdevices. *IEEE Transactions of Computer-Aided Design of Integrated Circuits and Systems*, 15(71), July 1996.

[23] Ted J. Hubbard. *MEMS Design.* Ph.D. thesis, Caltech, 1994.

[24] T.J. Hubbard and E.K. Antonsson. Emergent faces in crystal etching. *Journal of Microelectromechanical Systems*, 3(1):19–28, 1994.

[25] T.J. Hubbard and E.K. Antonsson. Cellular automata modeling in mems design. *Sensors and Materials*, 9(7):437–448, 1997.

[26] R.J. Jaccodine. Use of modified free energy theorems to predict equilibrium growing and etching shapes. *J. Appl. Phys.*, 33(8):2643–2647, August 1962.

[27] W. Kern and C.S. Deckert. Chemical etching. In J.L. Vossen and W. Kern, editors, *Thin Film Processes.* Academic Press, Orlando, 1978.

[28] E. S. Kim, R. Muller, and R. Hijab. Front to backside alignment using resist patterned etch control and one etching step. *Journal of Microelectomechanical Systems*, 1(2):95–100, June 1992.

[29] A. Koide, K. Sato, and S. Tanaka. Simulation of two-dimensional etch profile of silicon during orientation-dependent anisotropic etching. In *Transducers '91*, pages 216–220, Institute of Electrical Engineers, 1991.

[30] George M. Koppelman. Oyster, a three-dimensional structural simulator for microelectromechanical design. *Sensors and Actuators A*, 20(1):179–185, 1989.

[31] Hui Li and Erik K. Antonsson. Mask layout synthesis through an evolutionary algorithm. In *Proceedings of the International Conference on Modeling and Simulation of Microsystems*, 1999.

[32] M. Madou. *Fundementals of Microfabrication.* CRC Press, Boca Raton, 1997.

[33] F. Maseeh, R. Harris, and S. Senturia. A CAD architecture for MEMS. In *Transducers 90*, pages 44–49, Institute of Electrical Engineers, 1990.

[34] G. K. Mayer, H. L. Offereins, H. Sandmaier, and K. Kuhl. Fabrication of non-underetched convex corners in anisotropic etching of (100)-silicon in aqueous KOH with respect to novel micromechanic elements. *J. Electrochem. Soc.*, 137(12):3947–3951, December 1990.

[35] W.M. Moreau. *Semiconductor Lithography.* Plenum Press, New York, 1988.

[36] B. Nikpour, L.M. Landsberger, and T.J. Hubbard. Concave corner compensation between vertical (010)-(001) planes anisotropically etched in si(100). *Sensors and Actuators A-Physical*, 66(1–3):299–307, April 1998.

[37] H. L. Offeriens, K. Kuhl, and H. Sandmaier. Methods for the fabrication of convex corners in anisotropic etching of (100) silicon in aqueous KOH. *Sensors and Actuators*, A25-27:9–13, 1991.

[38] Kurt E. Petersen. Silicon as a mechanical material. *Proceedings of the IEEE*, 79(8):420–457, May 1982.

[39] W.G. Pfann. Improvement of semiconducting devices by elastic strain. *Solid State Electron.*, 3:261–267, 1961.

[40] F. Pourahmadi and J. Twerdok. Modeling micromachined sensors with finite elements. *Machine Design*, pages 44–60, July 1990.

[41] B. Puers and W. Sansen. Compensation structures for convex corner micromachining in silicon. *Sensors and Actuators*, A21-23:1036–1041, 1990.

[42] H.R. Robbins and B. Schwartz. Chemical etching of silicon-i. the system, hf, hno3 and h2o. *Journal of the Electrochemical Society*, 106:505–508, 1959.

[43] H.R. Robbins and B. Schwartz. Chemical etching of silicon-ii. the system, hf, hno3, h2o, and hc2c3o2. *Journal of the Electrochemical Society*, 107:108–111, 1960.

[44] K. Sangwal. *Etching of Crystals.* Elsevier Science Publishers B.V., New York, 1987.

[45] B. Schwartz and H.R. Robbins. Chemical etching of silicon-iii. a temperature study in the acid system. *Journal of the Electrochemical Society*, 108:365–372, 1961.

[46] B. Schwartz and H.R. Robbins. Chemical etching of silicon-iv. etching technology. *Journal of the Electrochemical Society*, 123:1903–1909, 1976.

[47] H. U. Schwarzenbach, J. G. Korvink, M. Roos, G. Sartoris, and E. Anderheggen. A micro electro mechanical CAD extension for SESES. *J. of Micromechanics and Microengineering*, 3:118–122, 1993.

[48] H. Seidel, L. Csepregi, A. Heuberger, and H. Baumgartel. Anisotropic etching of crystaline silicon in alkaline solutions. *J. Electrochem. Soc.*, 137:3613–3631, 1990.

[49] S. D. Senturia. Cad challenges for microsensors, microactuators, and microsystems. *Proceedings of the IEEE*, 86(8):1611–1626, 1998.

[50] S. D. Senturia, R. M. Harris, B. P. Johnson, S. Kim, M. A. Shulman, and J. K. White. A computer-aided design system for microelectromechanical systems (MEMCAD). *Journal of Microelectomechanical Systems*, 1:3–13, March 1992.

[51] C. H. Sequin. Computer simulation of anisotropic crystal etching. In *Transducers '91*, pages 801–806, San Francisco, CA, USA, 1991.

[52] James A. Sethian and David Adalsteinsson. An overview of level set methods for etching, deposition, and lithography development. *IEEE Transactions of Semiconductor Manufacturing*, 10(1), 1997.

[53] Jeff So. Convex hull java applet. *http://www.cs.uleth.ca/students/soj/*, 1996.

[54] S.M. Sze. *Semiconductor Devices: Physics and Technology.* New York, 1983.

[55] C.R. Tellier. Etching of semiconductor cubic crystals:determination of the dissolution slowness surfaces. *J. Crystal Growth*, 100:515–526, 1990.

[56] O. Than and S. Buttgenbach. Simulation of anisotropic chemical etching of crystalline silicon using a cellular automata model. *Sensors and Actuators A-Physical*, 45:85–89, 1994.

[57] L.F. Thompson, C.G. Willson, and M.J. Bowden. *Introduction to Microlithography.* American Chemical Society, Washington, D.C., 1994.

[58] Kenny K. H. Toh, Andrew R. Neureuther, and Edward W. Scheckler. Algorithms for simulation of three-dimensional etching. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 13(5), 1994.

[59] O.N. Tufte, P.W. Chapman, and D. Long. Silicon diffused-element piezoresistive diaphrams. *J. Appl. Phys.*, 33(3322), 1962.

[60] D.R. Turner. Electropolishing silicon. *Journal of the Electrochemical Society*, 105:402–408, 1958.

[61] T.W.Harris. *Chemical Milling.* Clarendon Press, Oxford, 1976.

[62] A. Uhlir. Electrolytic shaping of germanium and silicon. Technical report, Bell System Tech. J., 1956.

[63] X. Wu and W. H. Ko. Compensating corner undercutting in anisotropic etching of (100) silicon. *Sensors and Actuators*, 18:207–215, 1989.

[64] Xian-Ping Wu and Wen H. Ko. Compensating corner undercutting in anisotropic etching of (100) silicon. *Sensors and Actuators A*, 18:207–215, 1989.

[65] Q.X. Zhang, L.T. Liu, and Z.J. Li. A new approach to convex corner compensation for anisotropic etching of (100) si in koh. *Sensors and Actuators A*, 56(3):251–254, 1996.

[66] D. Zielke and J. Fruhauf. Determination of rates for orientation-dependent etching. *Sensors and Actuators A*, 48:151–156, 1995.