ABSTRACT

Title of Dissertation: STUDY OF WET-CHEMICALLY-PREPARED HYDROGEN-TERMINATED SILICON (111) SURFACES AND A NOVEL IMPLEMENTATION OF A HIGH-RESOLUTION INTERFEROMETER

Hui Zhou, Doctor of Philosophy, 2003

Dissertation directed by: Dr. Richard Silver
Chemical Physics Program

This thesis summarizes my graduate study under the National Institute of Standards and Technology (NIST) Atom-Based Dimensional Metrology Project, in which we are developing methods for measuring sub-micrometer dimensions including directly counting atom spacings on a silicon-surface lattice.

Atomically flat, hydrogen-terminated Si(111) surfaces are prepared using wet chemistry. The surface morphology after the wet-chemistry preparation was found to be dependent on both the initial etching time and wafer miscut. These two factors have been neglected in literature. To produce a morphology of uniform, long-range steps and terraces, the miscut angle has to be larger than a certain angle. The development and dynamics of the surface morphology was

explained by preferential etching. A kinetic Monte-Carlo simulation was used to quantitatively study some of the key aspects of the surface-morphology evolution, such as step flow, pit expansion, and step–pit collision.

The hydrogen-terminated silicon surfaces prepared using wet-chemical etching method were used as substrates to create nanometer-scale patterns using a scanning tunneling microscope (STM)-probe-induced surface modification in both ultra-high vacuum (UHV) and low-vacuum environments. Patterns created in UHV have linewidths below 10 nm, while patterns created in low vacuum had a minimum linewidth of nominally 20 nm. The pattern created in a low vacuum environment was further processed using SF_6 reactive-ion etching, resulting in patterns whose aspect ratio had increased more than 5 times.

To enable accurate measurement of atom spacings, a Michelson interferometer of novel design was implemented in this research, based on the principle that during operation, the interference-fringe signal is locked at a zero point by tuning the laser frequency, thus transferring the displacement measurement into a laser-frequency measurement and greatly increasing the measurement resolution. The interferometer is designed to be integrated into an ultra-high-vacuum scanning tunneling microscope for atom-resolved measurements. This unique implementation achieves a nominal resolution of sub-angstrom. In this thesis, the principles of the interferometer design and the uncertainty budget of the interferometer are discussed.

STUDY OF WET-CHEMICALLY-PREPARED HYDROGEN-TERMINATED SILICON (111) SURFACES AND A NOVEL IMPLEMENTATION OF A HIGH-RESOLUTION INTERFEROMETER

by

Hui Zhou

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2003

Advisory Committee:

Prof. Ellen D. Williams, Chair
Prof. Janice Reutt-Robey
Prof. Peter Kofinas
Prof. Sheryl H. Ehrman
Prof. Gottlieb S. Oehrlein
Dr. Richard M. Silver

DEDICATION

To My Wife, Xiaoyi Zhang

ACKNOWLEDGMENTS

I would like to start by expressing my eternal gratitude to my thesis advisor, Dr. Richard Silver, for offering me the opportunity to do my graduate research in the rich environment of a national laboratory and for giving me the support and encouragement that enabled me to get through my thesis research. There have been quite a few rough times throughout my study, and I will always remember his persistence during those times.

An extra special thanks also goes to Dr. Ellen Williams for taking the responsibility of being my academic advisor. Given her active leadership and extensive involvement in a number of research projects, I am in debt of the time she has spent in carefully going over my reports and thesis. Her comments and insights have proved to be very valuable for my research, and without her guidance, this thesis would not have been possible.

Much gratitude goes to two people without whose help I could not possibly have completed this work: Joseph Fu and Bradley Damazo. Joseph has spent a great deal of time with me doing the experiments, discussing ideas, and working through problems, to the extent that there is so many places that in my thesis I have to mention his name again and again. And Bradley, who is very thoughtful

and kind, has always been my first source of help whenever I have written awkward text that needed to be revised, encountered problems in doing my research, or needed advice of some kind, whether inside or outside of NIST.

To John Kramar for going over my presentations, to Vincent Scheuerman for initially teaching me to operate the lab facilities, to George Witzgall for collaborating on nano-patterning the silicon surface, to Zolt Radi for engaging in numerous interesting discussions and for sharing his experience in making nano-probes, to Satoshi Gonda for the year we spent in the lab together, and to Russ Hajdaj for offering his assistance in using the micro-fab facilities: Thank you!

Finally, I am forever indebted to my parents, Zhou Jinlin and Gu Caifen, for you have made me who I am today. To my in-laws, Zhang Tiyun and Hu Yunchun, who helped out earlier this year by taking care of my baby son, I am very appreciative. To my wife, Xiaoyi Zhang, I am forever grateful for your support. And to my son, Harry: when you are old enough to read this, I want you to know that you were always in my heart as I was writing it.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction: Atom-Based Dimensional Metrology Project

1.1 The era of nanotechnology

In 1965, Gordon Moore observed that the number of transistors placed on a single integrated circuit was growingly exponentially with time [1]. Amazingly, this observation still holds true today and is called Moore's Law. Today, a Pentium 4 processor has 55,000,000 transistors crammed onto a single chip [2], and it is manufactured using photolithography technology with a critical dimension of 130 nm. Moore's Law predicts that this size will shrink by a factor of 10 every 6 to 7 years, and thus in the near future the semiconductor industry will focus on dimensions in the nanometer range.

The word *nanotechnology* has become very popular recently. It is defined by the National Nanotechnology Initiative (NNI) as "research and technology development at the atomic, molecular or macromolecular levels, in the length

scale of approximately 1–100 nanometer range, to provide a fundamental understanding of phenomena and materials at the nanoscale and to create and use structures, devices and systems that have novel properties and functions because of their small and/or intermediate size" [3]. Two major branches of research exist and should be distinguished. One is the continuing reduction in the critical dimensions of current or conventional technology. For example, continued improvements in lithography have resulted in line widths that are down to 10 nm [4]. If we view this branch of research as approaching the atomic limit from the top down, then the other branch, molecular nanotechnology, can be seen as an attempt to reach this limit from the bottom up: to build devices such as nanowires [5] and molecular electronics [6] by directly assembling atoms. There is no question that molecular nanotechnology will influence our lives in the future in many ways; however, it is still in its infancy and requires time to mature. In the next decade, advances in the conventional branch of nanotechnology will continue to lead the industry.

Nanotechnology requires nanometrology. This is true for both branches of nanotechnology. Currently, the ability to measure feature positions or critical dimensions to sub-nanometer accuracy is a pressing problem in some cutting-edge manufacturing environments. In semiconductor manufacturing, for example, the gate-length dimensions need to be measured accurately to 3 nm for devices that are of 150 nm in size, as stated in the International Technology Roadmap for Semiconductors (ITRS) [7]. The need for sub-micron reference standards and

methods for making submicron measurements with tolerances approaching the atomic size will become more and more demanding as the industry continues to reduce the dimensions of critical features. In order to build devices atom by atom in the future, it will first be necessary to measure devices atom by atom.

To fulfill its mission "to develop and promote measurement standards and technology to enhance productivity, facilitate trade and improve the quality of life" [8], the National Institute of Standards and Technology (NIST) has launched the *Atom-Based Dimensional Metrology Project* [9] to develop nanometer- to submicron-dimensional standards based on the counting of atoms in a crystal-surface lattice. The following gives an overview of this project, of which the research presented in this thesis is a part.

1.2 Goals and strategies

The long-term goal of this project is to produce reference standards with features or dimensions established on the basis of surface atomic lattices by means of atom counting via scanning tunneling microscopy (STM). To accomplish this task, it is necessary to develop fabrication methods to prepare 3D patterned surfaces suitable for atom counting and to develop a methodology for making calibrated high-resolution measurements of surface features. Both address the current difficulty in the semiconductor industry and are listed as intermediate goals of the atom-based dimensional metrology project.

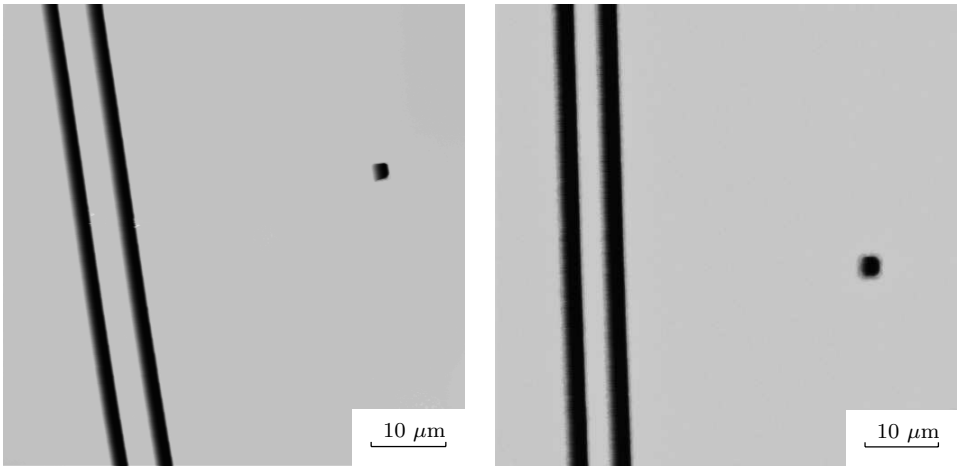
There are three strategies we can use to achieve these goals. In the first strategy, a set of carefully designed test structures are first fabricated on a substrate using conventional photolithography methods or direct e-beam writing. This is followed by surface processing, to render the surfaces atomically ordered and measurable by a high-resolution STM. To actually "count" the atoms, a set of atomically resolved STM images are acquired across the features of interest. A variety of methods can be used to determine the number of atomic spacings across the feature. One method might utilize a computer-generated template, which is first laid over the image, and then the atom spacings are counted in an automated fashion. After the UHV STM measurement, the samples are stabilized for use at atmospheric pressure by either *in situ* thermal oxidation or by exposure to the ambient environment in which a native oxide is allowed to grow. During the stabilization period, the surface-oxide thickness typically would change by a few nanometers but would otherwise appear to be dimensionally stable. The test structures that are so designed should be dimensionally measurable by atomic force microscopy (AFM), scanning electron microscopy (SEM), or optical microscopy, and can be used to directly calibrate the respective instruments.

The surface processing is the weakest link in this strategy. Conventional techniques such as high-temperature annealing work well on a homogeneous surface, but they are less effective on surfaces with prefabricated features. The sharp edges typically required in metrologically meaningful structures are thermodynamically unstable and tend to become deformed as their energy is

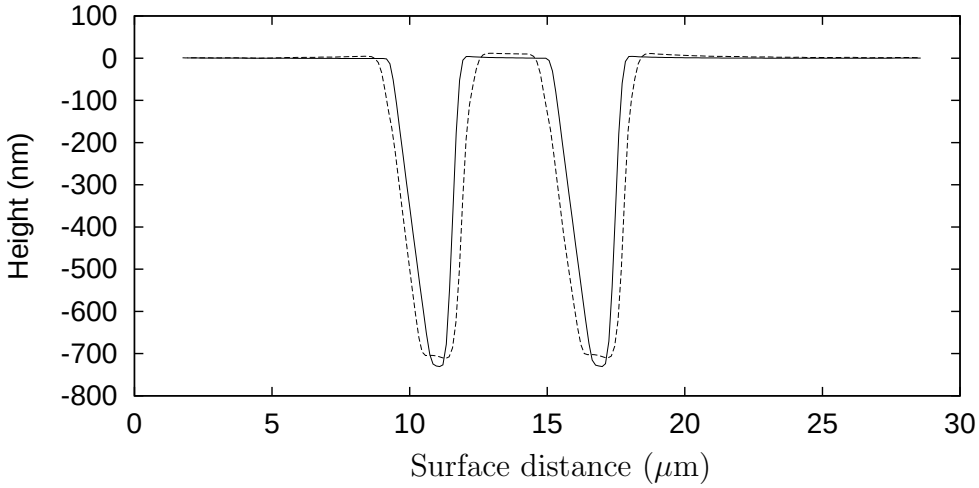
lowered during the annealing process. Figure 1.1 shows a line-structure before and after the anealing process showing the significant effects of edge deformation. A more systematic study by E. D. Fu et. al. also revealed severe structure decay after high temperature annealing process [10]. While the edge deformation might be tolerated for pitch measurements, where only the positions of the centers of lines need to be determined, it typically renders line-width structures useless.

For silicon (111) surfaces, wet-chemistry methods [11] can be used to achieve long-range atomically ordered surfaces. Figure 1.2 shows a line-structure before and after a 5-minute etching in NH_4F. During this process, the edges of the structure are etched by the chemical etchant at a higher rate than flat terraces are. The surface structure still suffers minor deformation, thereby making it more difficult to achieve atomically flat surfaces near the edges than on the plateaus. However, the extent of structure damage by the wet chemistry process is much less than that by the high temperature annealing process, making the wet chemistry process more appealing in this application.

The deformation of structures during the surface processing contributes the major uncertainty in measuring dimensions via the strategy described above. In the second strategy, perfect atomically ordered surfaces are obtained on flat substrates, and then patterns are fabricated using scanning-probe surface modifications. Since the patterns are fabricated on top of the surface atomic lattice, the dimensions of the pattern can be determined to within the accuracy of atomic spacings. The patterns fabricated by a scanning probe typically have a

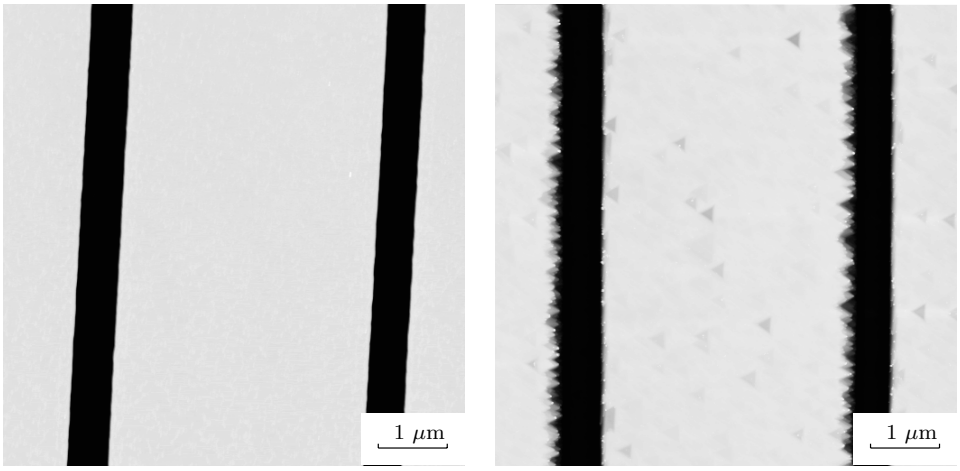


(a) (b)

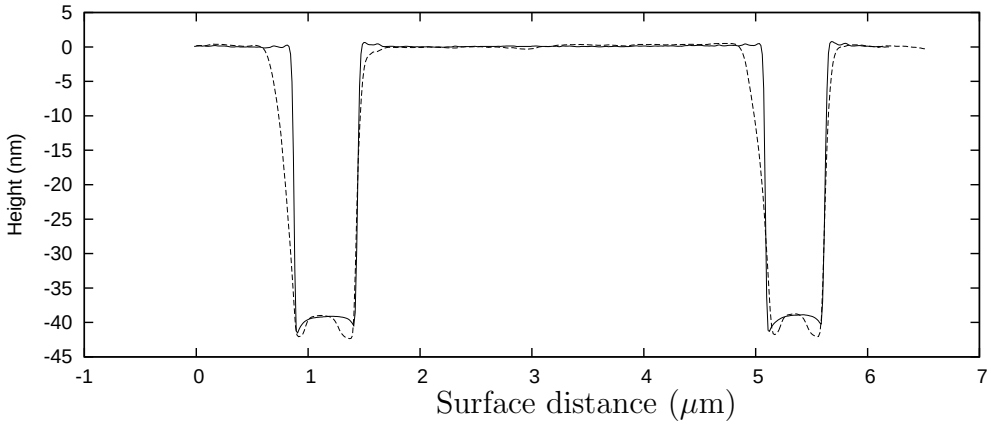


(c)

Figure 1.1. Two AFM images of a surface taken in air showing two lines of the repetitive structure, (a) before surface treatment and (b) after 10 seconds of UHV heating at 1000°. The cross-section profile of the images is shown in (c). Note the deformation of the structure edges. The UHV high temperature annealing process was performed by Carsten Jensen, and the AFM measurements were performed by Joseph Fu.



(a) (b)



(c)

Figure 1.2. Two AFM images of a surface taken in air showing two lines of the repetitive structure, (a) before surface treatment and (b) after 5 minutes of etching with NH_4F. The formation of etching pits on the top of the structure is an indication of formation of atomic flat terraces. The cross-section profile of the images is shown in (c). Note that the deformation of the structure edges is much less than that which resulted from the thermal annealing process. AFM measurements performed by Joseph Fu of NIST.

feature height of only a few nanometers and must be transferred into features with more contrast in order to be measurable using AFM, SEM, or optical microscopy. In this strategy, the pattern-transfer process is expected to introduce the most uncertainty into the measurements of the dimensions. One candidate for this method is using wet-chemically-prepared Si (111) surfaces followed by UHV STM surface modification [12] through an electron-stimulated desorption (ESD) process [13].

To further support the atom-based metrology, as a third strategy, a high-resolution interferometer is being implemented in this research based on a unique design; it is miniature in size and will be integrated into a UHV STM. This new tool will enable us to directly measure the dimensions of structures with sub-angstrom accuracy. Although it is originally designed to measure the surface atomic spacings and only operate with a small measurement range (a few nanometers), it is possible to extend to directly measure structure dimensions of larger sizes, up to a few hundred nanometers. This provides an alternative strategy to achieve the goals of this project. Although with this strategy, the measurement is not directly based on atom counting, it still aims at or beyond atomic accuracy.

1.3 Project components and key results of this thesis research

In the following I will give a short introduction to some of the components in the atom-based metrology project that my thesis research involves. Most of the progress that I summarised here were largely results of this thesis research, and will be discussed in detail in the rest of the thesis.

1.3.1 Preparation of hydrogen-terminated Si (111) surfaces using wet chemistry

In order to directly count surface atoms, atom-countable surfaces are required. The preparation of long-range atomically ordered surfaces has been one of the major tasks. For silicon surfaces, two methods can be used for this purpose: the conventional high-temperature annealing process in UHV and the wet-chemical process in the ambient environment.

If the sole purpose is to obtain long-range atomically ordered surfaces, a high-temperature annealing process is the method of choice, as the UHV environment renders it relatively easier to keep the surfaces free from contamination. The process produces a 7×7 reconstruction on Si (111) [14] and a 2×1 reconstruction on Si (100) [15], both of which have been extensively studied. Figure 1.3 shows one of the images of reconstructed Si (111) surface acquired

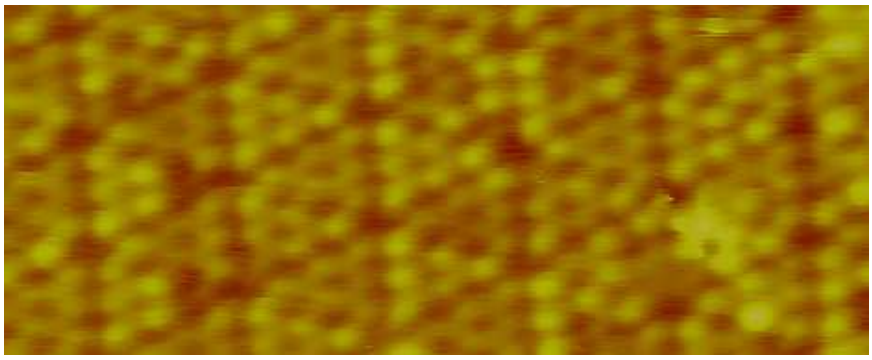


Figure 1.3. An image of Si (111) 7×7 reconstruction.

with the our UHV STM. However, the structural deformation of the critical features is very difficult to work around and forced us to look for a more "gentle" method to produce long-range atomically ordered surfaces. The use of hydrogen-terminated Si (111) surfaces prepared using wet chemistry is one such alternative method. The wet-chemistry processing is performed at room temperature, in a temperature region in which the surface atoms are assumed to be "frozen" in their lattice positions. The atomically ordered surfaces are obtained by kinetically removing most "non-flat" surface-atom sites, which typically requires removal of only one or two surface monolayers. Thus the prefabricated structures suffer much less damage than under thermal processing.

Hydrogen-terminated surfaces also provide an ideal substrate for performing scanning-probe-induced surface modifications [16, 17, 18]. It should be noted that hydrogen-terminated surfaces can also be obtained by depositing hydrogen

atoms on surfaces prepared through high-temperature annealing; however, exploration of the application of STM-based surface modification requires the ability to locate the UHV-STM-produced patterns with another instrument such as an AFM. Without the ability to place a mark which persists on the surface, searching for the UHV-STM-fabricated patterns on a sample area of nominally a square centimeter is like looking for a needle in the haystack. While it is essentially impossible to embed such markings in such a way that they will survive the high-temperature annealing process, this is not a problem for the wet-chemistry method. Because of these advantages of using wet chemistry, a great amount of effort has been put into the development of a robust method of producing long-range atomically ordered surfaces via wet chemistry.

By treating the Si(111) surfaces in 40% NH_4F solution with the dissolved oxygen depleted, it has been reported by other researchers that large terreaces and uniform step distributes result [19], with atomically flat terraces [11]. These surfaces are ideal for contructing nano-patterns. Although it was reported in 1993 that drastically different morphologies resulted from different surface miscuts [20], it is still often neglected in recent publications. During our initial study, we found it difficult to reporduce a morphology following the previous study in literature. We then realized that we were using samples that have smaller miscut angles than that commonly used in literature. After a systematic study, we found that the final surface morphologies greatly dependent on both the surface miscut and the initial etch time. To obtain flat surfaces with uniform

steps and terraces, the sample miscut angles have to be above a certain value and the samples have to be etched in the final solution long enough to reach a steady state; moreover, the time required to reach a steady state is also dependent on the sample miscut angle. The step-flow mechanism [21] was revisited to discuss the various morphology development.

The most successful theoretical treatment for the etching of Si(111) in NH_4F has been the kinetic Monte-Carlo simulation by M. A. Hines [22]. The previous work by Hines et al. has obtained a set of optimum site specific etch rates that simulate the steady state morphology of samples with miscut 0.35° either toward $< 11\bar{2} >$ or $< \overline{11}2 >$ directions. We further improved the simulation method by incorporating a consistent time scale during the simulation to theoretically explore the surface morphology development, such the step-flow rate for various miscut configuration, the non-linear pit-growth curve, and the formation of stacking pits with non-defect origins. In order to simulate our experiment data, we found it necessary to consider the pit initiation due to the crystal defects and dopant. By modifying the algorithm to incorporate the defect concept and adjusting the parameters correspondingly, the simulation successfully produced satisfactory agreement with our experiments. The theoretical simulation study helped us gain further insight into the mechanism of Si(111) etching in NH_4F.

1.3.2 Surface modification on a hydrogen-terminated Si surface

Hydrogen-terminated silicon surfaces are relatively stable against oxidation, even in air, for a long time. However, by employing the nano-positioning capabilities and high current throughput of a scanning probe microscope (SPM), we were able to selectively desorb the hydrogen termination and achieve fabrication of high-resolution patterns. Similar processes have been demonstrated in UHV, albeit possibly by a different mechanism. The research in this project has focused on the UHV process and has demonstrated the capability of fabricating line patterns with critical dimensions of ≈ 10 nm.

To access the patterns created by an STM created in a UHV environment with another instrument, such as AFM, SEM or optical microscopes, requires the patterns to be transferred into features with more surface contrast and invariant against oxidation in an ambient environment. For example, in order to use an optical microscope to measure the patterns, it generally requires at least 50 nm surface contrast. The difficulty in pattern transfer has been a major barrier to the applications of this nano-patterning technology. Conventionally, a UHV high-resolution STM does not have a macro vision system, so that any pattern created or accessed by such a system is like needle in a haystack, and is therefore often impossible to locate with other instruments. This prevents the study of possible pattern-transfer techniques and makes it impossible to use other

instruments such as AFM, SEM, or optical tools in the study of the patterning process, which is very important in gaining more insights into the mechanism of the scanning-probe-induced surface modification and in producing meaningful metrological reference standards.

The results of our experiments in UHV STM modification of hydrogen-terminated Si (111) surfaces prepared using wet chemistry and similar experiments in a low-vacuum environment are discussed. The low-vacuum experiments were performed to demonstrate the possibility of transfer of patterns created in UHV. Without hampering the vacuum in the UHV experiments, we were able to use a camera system in the vacuum chamber to locate the position of the STM tip on the sample. This enabled us to locate the pattern created in vacuum with an AFM operated in air, and hence to experiment with pattern transfer and to demonstrate the possibility of transfer of a pattern created under UHV conditions.

1.3.3 High-resolution interferometer system

The main concept of the project is based on the surface atomic lattice, which nominally has atomic resolution. The surface lattice constant is taken directly from the bulk-crystal lattice constant obtained through X-ray diffraction experiments. To partially support the concept of this project, we have designed an interferometer system to be integrated into a high-resolution UHV STM unit.

The interferometer has a design accuracy in the sub-angstrom range. By imaging an atomically ordered surface, we can measure the surface atomic lattice directly. This interferometer has been implemented, and its resolution has been demonstrated on a graphite surface with a previous prototype [23].

This interferometer system will enable us to directly measure surface-feature dimensions with atomic resolution, without the need to count atoms. Because of its high-resolution design, the measurement range for single-mode operation is limited to ≈ 100 nm. To carry out the task of measuring a typical feature, the range has to be extended by mode hopping, which will use this interferometer in a more general application. In this thesis research, we completed the laser frequency measurement module, developed the data acquisition and feedback control software, and bench tested the whole system to further estimate the uncertainty budget.

1.4 Organization of the thesis

The remaining parts of this thesis are organized as follows.

Chapter 2 consists of a discussion of the considerations that arise in applying the chemical procedures used to produce ideally mono-hydride-terminated Si (111) surfaces; the main results that show the dependence of the surface morphology on the etching time and sample miscut angle are presented there. A preferential etching model is used to explain the morphology evolution during

etching and the final morphology dependence on the miscut. In chapter 3, our attempt to use kinetic Monte-Carlo calculations to simulate some of the aspects of the chemical etching is described. The kinetic Monte-Carlo simulation was originally used by Hines [24] to extract atom-specific reaction rates. In this chapter, our application of a set of optimum parameters obtained by Hines to study the surface morphology evolution is discussed. Chapter 4 discusses the modification of the original kinetic Monte-Carlo simulation and its application to directly study the surface morphologies we observed in our experiments. Chapter 5 contains a discussion of our study on using STM-probe-stimulated hydrogen-terminated Si (111) surface modification to build nano-structures and the possible mechanism for the patterning process. The principle and implementation of our new interferometer setup and the determination of its uncertainty budget are discussed in chapter 6.

Chapter 2

Hydrogen-Terminated Si (111) Surface Preparation Using Wet Chemistry and Its Morphology Dependence on Wafer Miscut

2.1 Introduction

Hydrogen-terminated silicon surfaces were widely used in industry before the nature of the surfaces was understood. In particular, sequential peroxide-solution-based chemical oxidation followed by HF etching (RCA cleaning [25]) has been remarkably successful in preparing ultra-clean surfaces for subsequent thermal oxidation via metal-oxide-semiconductor (MOS) technology. The remarkable interfacial properties of the surfaces after HF treatment were later explained by hydrogen termination [26]. The atomically flat, unreconstructed Si (111) surfaces can be terminated by a single layer of mono-hydride (Figure 2.1). On an ideally mono-hydride-terminated surface, all

the surface dangling bonds are passivated by Si–H bonds, and the structures suffer minimal stress. Hydrogen-terminated silicon surfaces were found to be resistant to contamination and oxidation under ambient environmental conditions [27, 28]. One study of initial oxidation on hydrogen-terminated Si (111) surfaces indicated that nucleation of the oxide is still negligible after 7 hours in moist air [29]. It is expected to be more stable in dry air, since water is found to be crucial in the oxidation process [30]. The binding energy of the Si–H bond is greater than that of the Si–Si bond. Studies in quantum chemistry have shown that during the initial oxidation, interstitially absorbing the oxygen in the Si-Si bonds forms a more stable structure than surface Si–OH terminations [31]. This is supported by studies using infrared (IR) spectroscopy which found that large numbers of Si–H bonds remain on the surface after the native oxide has grown.

The simplest method to obtain a hydrogen-terminated Si surface is to first chemically oxidize the surface and then strip off the oxide with HF solution. The HF-treated Si (111) surface is predominantly terminated by silicon mono-hydride, as indicated by IR spectroscopy [32]. A low-energy electron-diffraction (LEED) study indicates that the surface after only HF etching remains unreconstructed [33]. However, studies of the complexity of the IR spectra and STM measurements show that these surfaces are microscopically rough [32]. The uniformity of hydrogen termination on Si (111) surfaces is significantly improved by buffering the HF solution with NH_4F. By using the extremely buffered solution, 40% NH_4F, the treated Si (111) surfaces were ideally terminated by

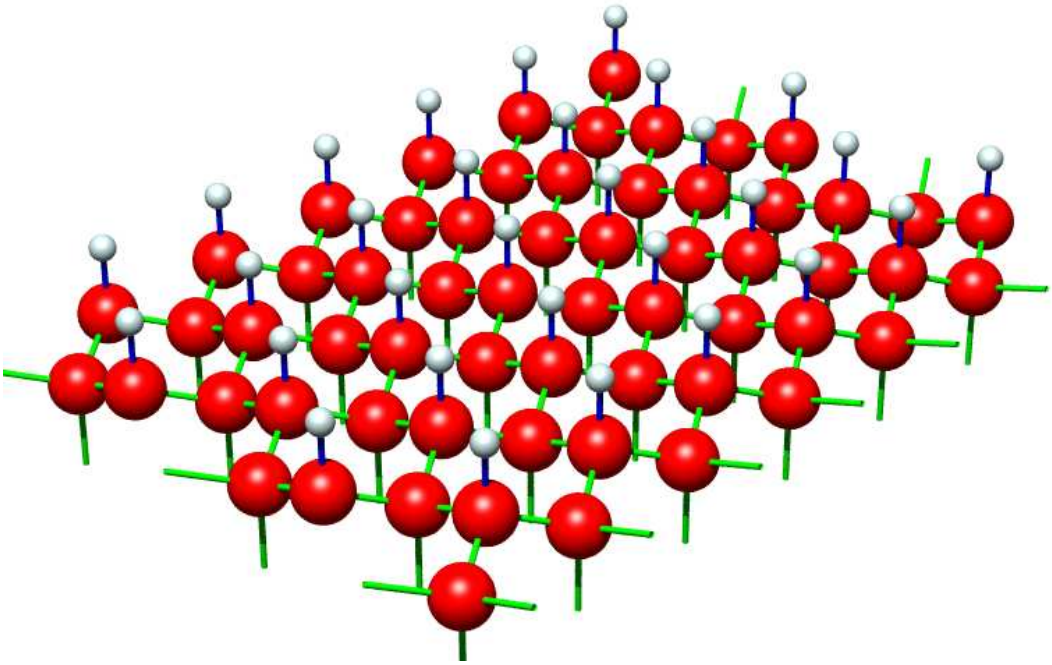


Figure 2.1. Illustration showing the structures of an unreconstructed hydrogen-terminated Si (111) surface. Each surface silicon atom has three bonds connected to the bulk silicon atoms and one bond that points upward, which is terminated by a mono-hydride and results in a stable molecular structure.

mono-hydride and were atomically ordered, as studied by Higashi in 1990 [28, 34, 35].

Unlike the etching of silicon in hydroxide solutions, the etching of Si(111) in NH_4F is very slow, and the etch rate is below the detection limit. For this type of etching, the surface morphology study has been proved to be an effective method to gain insights into the mechanism. In 1991, H. E. Hessel studied the surface morphology of Si(111) surfaces after 3 minutes of etching in HF solutions of different pH level by buffering with 40% NH_4F [21] The one etched with pH 8 solution was found to to form long range terraces and steps. They proposed a mechanism of step-flow versus pit corrosion which reasonably explains the formation of this morphology. In 1993, G. J. Pietsch noticed that the etching in NH_4F produced drastically different morphology for Si(111) samples with different miscuts [20]. The surface of the sample with 0.02° miscut angle was dominated by multiple level of etch pits after 4 minutes of etching, while the one with 0.5° miscut angle developed into long range steps and terraces.

For quite some time, the surface morphologies of Si(111) processed with 40% NH_4F found in the literature have been inconsistent. The inconsistency is partially due to the complexity of the wet-chemistry process. Unlike experiments under ultra-high-vacuum conditions, there are numerous uncontrolled or difficult to control factors in the wet-chemistry process, and each neglected factor can possibly play an important role in determining the final surface morphology. In 1997, Wade [19] discovered that dissolved oxygen in the etching solution, which is

commonly neglected because of its low concentration, plays an important role in increasing the pits during the etching. By suppressing the dissolved oxygen concentration, they reported that the Si(111) surfaces developed into a morphology with uniformly distributed steps and terraces and with few pits formation. Following this research, H. Sakaue in 2000 reported that they were able to produce atomic-scale defect-free Si(111) surfaces at wafer size [11].

These fascinating results of wet chemical processing of Si(111) are very attractive to our atom-based dimensional project. However, we experienced difficulty in repeating the surface morphology results by following the literature, until we experimented with various surface miscut configurations. We realized the samples being used have smaller miscut angles than those which were commonly used in the literature. Miscut describes the misalignment of the surface normal to the interested crystallography direction. To fully describe a miscut, one needs to specify both the tilt angle (miscut angle) and the azimuth angle (mis-orientation angle). Assuming a surface with certain miscut is perfectly flattened, the surface morphology will have equally distributed straight steps and terraces, on which the terrace width or step spacing is determined from the miscut angle and the step orientation is determined from the misorientation angle. Miscut results from the uncertainty during the wafer sawing process, which is unavoidable and very difficult to control. Table 2.1 lists the wafer miscut information in recent publications. Most studies in the literature have focused on surfaces with relatively large miscut angles ($> 0.3^{\circ}$), resulting in high-density

steps and terraces. Because of our interest in obtaining large terraces, we experimented with very small miscut angles. Also, the studies in the literature often focused on the steady state morphology after the surface had been etched for a certain time period. Our studies indicate that both the wafer miscut and the initial etching time are very important in determining the surface morphology after etching. Similar results regarding the miscut angles were also reported by Pietsch [20]. In our study, the time dependent surface morphology for samples with different miscut configurations was obtained by trying to plot the complete picture of the surface morphology evolution under etching in 40% NH_4F.

Thanh have studied the use of NH_4F to treat Si (100) surfaces and found the process also produced a flat, di-hydride-terminated surface [37]. In their studies, long-range atomically ordered surfaces, however, were not found, and (111) facets started to form after prolonged etching.

It should be noted that there is an alternative way to prepare atomically ordered, hydrogen-terminated silicon surfaces [38]: the silicon surface first goes through a high-temperature annealing process, during which both the surface oxide and the contaminants are throughly desorbed and the surface reconstructs itself to lower its surface energy, resulting in a flat, atomically ordered surface. The reconstructed surface is still very reactive. By dosing with atomic hydrogen—commonly prepared by cracking molecular hydrogen with a hot tungsten filament placed near the silicon surface in UHV—a reconstructed surface with a mono-layer of hydrogen termination can be produced. This

Table 2.1. Wafer information and key morphology results in the literature

Author	**year**	**miscut**	**time**	**morphology**
Higashi [34]†	1991	N/A	4-6 min.	N/A
Hessel [21]	1991	0.17° *,**	3 min.	Long range steps, numerous pits
Pietsch [20]	1993	0.02°	4 min.	Multiple level pits with size up to 300 nm
		0.5° *		Long range steps, few pits, steps meandering with straight portions
		0.5° **		Long range steps, step hillocks
		4° *,**		Long range, homogeneous steps
Wade [19]‡	1997	0.3° *	4 min.	Uniformly distributed steps, steps zigzagged, very few pits
Flidr [36]‡	1999	0.35° *	30 min.	Long range steps, a few pits
		0.35° **	30 min.	Long range steps, very few pits, step hillocks
Sakaue [11]†‡	2000	1° *	6 min.	Uniformly distributed steps, no pits

†atomically resolved, ‡dissolved oxygen suppressed

*miscut toward $< 11\bar{2} >$, **miscut toward $< \overline{11}2 >$

method is very reliable in a good UHV environment; however, it still has the disadvantage that the surface structures are deformed, thereby limiting its usefulness for our project.

2.2 Chemistry of silicon etching

Wet-chemical etching on a silicon surface can typically be viewed as a two-step process: The etchant first oxidizes the silicon, and then the oxide is removed from the surface [39]. For NH_4F etching, the oxidation is accomplished by the following reactions:

$$Si + 4OH^- \rightarrow SiO_2 + 2H_2O + 4e^- \tag{2.1}$$

and

$$4H_2O + 4e^- \rightarrow 4OH^- + 2H_2 \uparrow \tag{2.2}$$

Overall, it is an OH^--catalyzed reaction:

$$Si + 2H_2O \rightarrow SiO_2 + 2H_2 \uparrow \tag{2.3}$$

Once the oxidation has been achieved, the silicon oxide is removed by HF:

$$SiO_2 + 6HF \rightarrow H_2SiF_6 + 2H_2O \tag{2.4}$$

A 40% NH_4F solution is slightly alkaline (with a pH of ≈ 8), and the OH^- concentration is very low, thus the oxidation is the step that limits the rate at which the entire etching process proceeds.

The presence of dissolved oxygen in an aqueous solution has the ability to oxidize the silicon:

$$Si + O_2 \rightarrow SiO_2. \tag{2.5}$$

The reaction 2.5 and reaction 2.3 overcome different reaction-energy barriers, the relative etch rates of different atomic sites are very different with each reaction. It is observed that high concentration of dissolved oxygen is responsible for accelerating etch-pit formation, which results in a rough surface morphology [19]. A quantitative study by Hines [40] confirms this results. By depleting the dissolved oxygen in the etching solution, the pit formation during etching is significantly suppressed, resulting in a more controlled surface morphology. A similar effect of dissolved oxygen was also found in the etching of Si(111) in 2.5% NH_3 soultion by H. Fukidome [41].

2.3 Wet-chemistry procedures

2.3.1 Pre-cleaning

Before the silicon surface is etched in NH_4F, it needs to be cleaned to ensure that it contains no contaminants that would affect the etching process. The pre-cleaning is done out of concern for the presence of two main types of contaminants: organic carbon and heavy-metal particles. Organic carbon forms random etch masks on the surface, thereby precluding homogeneous etching

across the surface. Heavy-metal particles are notorious for sticking to the silicon surface and forming defect sites which disturb the development of surface morphology during the etching.

The most commonly used method for cleaning silicon surfaces is the one developed at the RCA labs [42, 25]. The key components of the RCA cleaning process are based on the chemistry of hydrogen peroxide, which is a powerful oxidation agent. Hydrogen-peroxide-based solutions are very effective in decreasing contamination due to organic carbon.

The original RCA cleaning approach entailed using two solutions sequentially. The first solution, typically 5:1:1 parts by volume of H_2O:H_2O_2:NH_4OH (named SC-1, which stands for standard clean, solution 1) is especially effective in removing organic contaminants, because of the fact that H_2O_2 oxidizes the organic molecules more effectively in alkaline solutions. However, it significantly roughens the silicon surface as a result of the etching effect of the OH^- ions. Thus this process should be avoided in pre-cleaning procedures. The second solution, typically 5:1:1 parts by volume of H_2O:H_2O_2:HCl (named SC-2), is also effective against organic contamination. However, it was originally designed for the purpose of removing heavy-metal particles that arose because of the strong acidity of the solution. Other solutions sometimes used include 4:1 parts by volume of H_2SO_4:H_2O_2, 1:1 parts by volume of H_2SO_4:HNO_3, or just HNO_3, all of which are strong acidic oxidation agents that have a similar effect as the SC-2 solution. The use of ultraviolet (UV) radiation to decompose the surface organic

contamination has also been reported to achieve good effects in cleaning silicon surfaces [43].

There are many variations in the pre-cleaning procedures used by various research groups. Because of many uncontrolled factors and a lack of quantitative studies, it is difficult to tell which procedure is best. The sequence of procedures that we currently use, which are listed below in order of application, have routinely produced ideally hydrogen-terminated Si (111) surfaces.

1. Generally de-grease the samples by rinsing with acetone in an ultrasonic cleaner. At the start of any process, a silicon surface is usually heavily contaminated with organic substances. Rinsing in an effective de-greasing solvent will remove most of those contaminants and increase the effectiveness of the remaining cleaning procedures in this list.

2. Rinse with de-ionized (DI) water in an ultrasonic bath.

3. Immerse the samples in a 1:1 parts by volume H_2SO_4:HNO_3 solution at 60–80°C for 1 hour.

4. Rinse with DI water for 5 minutes in an ultrasonic cleaner.

5. Immerse the samples in a 2% HF solution for 2 minutes. Silicon surfaces that have been handled in air have a thin layer of native oxide. The H_2SO_4:HNO_3 cleaning will also grow a thin layer of chemical oxide. The use of a dilute HF solution removes this oxide layer and re-exposes the

silicon surfaces, thereby making the remaining cleaning procedures in this sequence more effective.

6. Briefly rinse the samples with DI water by dipping them in water for 10–30 seconds. In carrying out this procedure, we kept the duration of the rinsing to a minimum, because of the fact that pure water will slightly etch the hydrogen-terminated silicon surface and increase its roughness.

7. Immerse the samples in a 5:1:1 parts by volume H_2O:H_2O_2:HCl solution at 80°C for 10 minutes.

8. Rinse with DI water for 5 minutes in an ultrasonic cleaner, at which point the samples are ready for the final etching in 40% NH_4F.

2.3.2 Final etching

After the sample surfaces are chemically cleaned, they are dipped into 40% NH_4F solution for the final etching. Because of the roughening effect of dissolved oxygen, the NH_4F solution is bubbled with nitrogen or argon for at least 30 minutes to lower its concentration of dissolved oxygen. The etching time in NH_4F is dependent on the sample miscut conditions and the experimental requirements. For example, a surface with a miscut angle of larger than 0.1° is typically etched for 15 minutes. At the completion of the etching, the samples are briefly rinsed by being dipped in DI water for a few seconds and then blown dry with high-purity nitrogen gas.

The surfaces that result from etching in NH_4F are hydrophobic. The etching solution rinses off the surfaces easily, resulting in atomically clean surfaces as observed under AFM. Occasionally however, we observe that particles of size anywhere from a few nanometers to 50 nm are produced (Figure 2.2). These particles seem to be formed after the samples have been taken out of the etching solution, because they have no apparent effect on the steps and terraces. It is commonly found that these particles are formed after a sample has been etched in NH_4F for an extended period (more than 45 minutes) or when the final brief rinsing is eliminated. While the cause of formation of these particles is still under investigation, considering we are using saturated NH_4F solution (40%), it is likely that they are formed by crystallization of NH_4F and that the final brief rinsing helps to reduce the remaining NH_4F on the surface. Because water expedites surface oxidation, the rinsing time is kept short, to avoid the roughness incurred by the initial oxidation.

2.4 Preferential etching dynamics on Si (111)

2.4.1 Nomenclature

It is beneficial to define specific nomenclature with respect to the Si (111) surface miscut and step orientations for use throughout this thesis. Figure 2.3 is a schematic of these orientations. A miscut angle specifies the angle between the

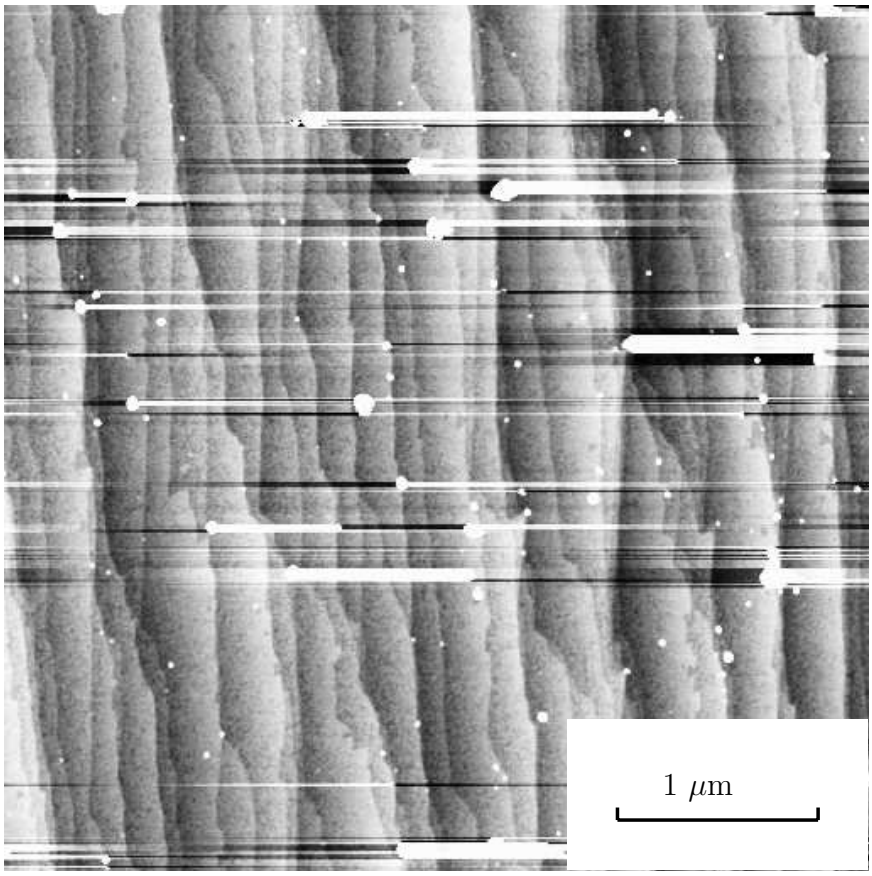
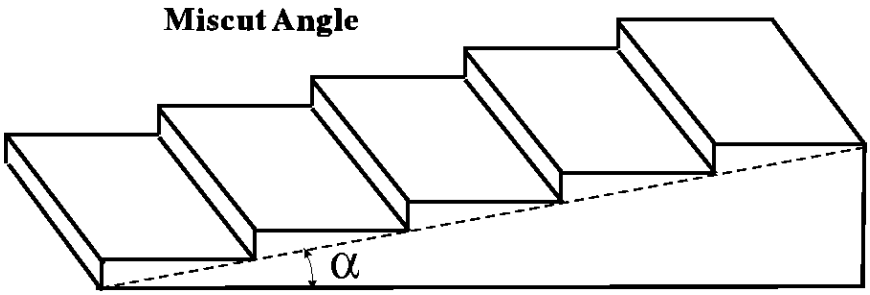


Figure 2.2. AFM image of a sample surface after NH_4F etching, showing particles all over the surface. The horizontal streaks stem from the AFM scanning process. The image shows no apparent correlation between the particles and the steps. The particles are believed to be formed by crystallization of NH_4F. The AFM measurement was performed by Joseph Fu of NIST.

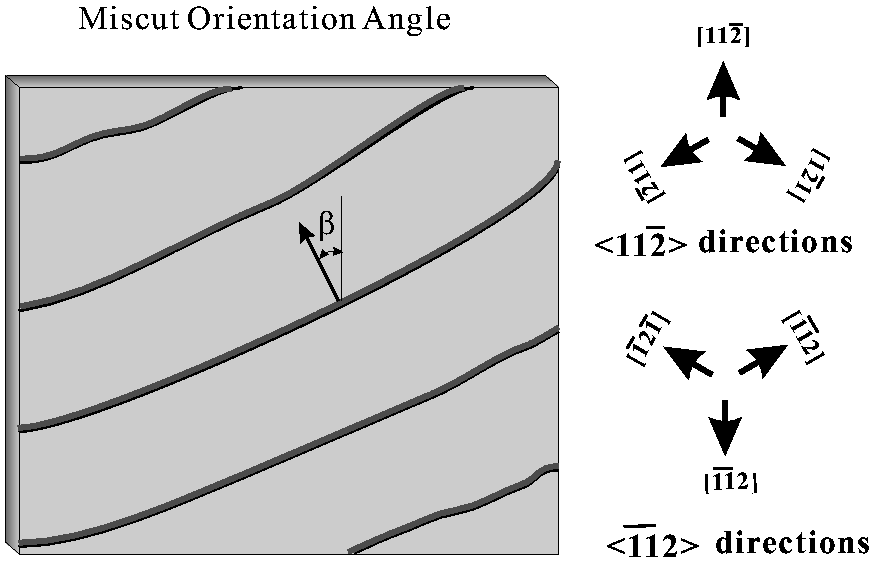
physical surface plane and its intended crystallography plane. For ideally flat surfaces, the miscut can be represented as a step-terrace staircase structure as in Figure 2.3a, where each terrace is identified by a crystallography plane and each step is a single atomic step with a constant height.

To fully describe a surface miscut, it is also necessary to specify its azimuthal orientation, as illustrated in figure 2.3b. For Si (111), the surface has a three-fold symmetry, resulting in three crystallographically indistinguishable directions — $[11\bar{2}]$, $[1\bar{2}1]$, and $[\bar{2}11]$. In this thesis, they are generally referred to as the $< 11\bar{2} >$ directions. Similarly, $[\overline{11}2]$, $[\bar{1}2\bar{1}]$, and $[2\overline{11}]$ are generally referred to as the $< \overline{11}2 >$ directions. Throughout this thesis, the azimuthal orientation of a miscut is specified by its angle between the $< 11\bar{2} >$ directions. This angle is referred to as the mis-orientation angle.

Given a certain miscut orientation, an ideally flat surface results in atomic steps with each step having a normal vector which is defined by the miscut azimuthal orientation. If the given miscut is exactly toward the $< 11\bar{2} >$ directions, the resulting steps will have their normal vectors pointing to the $< 11\bar{2} >$ directions, and described in this thesis as steps facing the $< 11\bar{2} >$ directions.



(a)



(b)

Figure 2.3. Illustrations showing the surface miscut orientations on Si (111). (a) illustrates the miscut angle. (b) illustrates the miscut orientation.

2.4.2 Typical surface morphologies

Figure 2.4 shows a typical surface morphology of Si (111) after NH_4F etching. The overall surface is very flat as all the terraces are flat Si(111) planes and all the steps are single atomic steps with heights of 0.3 nm. This is in contrast to a surface after a more aggressive etching such as KOH etching. Although vicinal Si(111) surfaces after KOH etching often displays a similar steps and terraces morphology [44], one should note that the steps resulting from KOH etching often are as tall as 200 nm, commonly referred to as macrosteps or ledges. The terraces between these ledges are not atomically flat.

Another feature of the morphology in figure 2.4 is the triangular etch pits. These etch pits are formed by single steps facing the directions of $[11\bar{2}]$, $[1\bar{2}1]$, and $[\bar{2}11]$. These etch pits should be distinguished from "macro etch pits" that are results of macro etching, which are much larger in size and enclosed by macro steps or facets. Macro etch pits are commonly believed to be formed from dislocations, while the single level etch pits result from the random etching of a terrace site of a perfect crystal plane. Both the single level etch pits resulted from NH_4F etching and the macro etch pits resulted from KOH rapid etching share the same symmetry, which suggesting that both may have the same preferential etching properties.

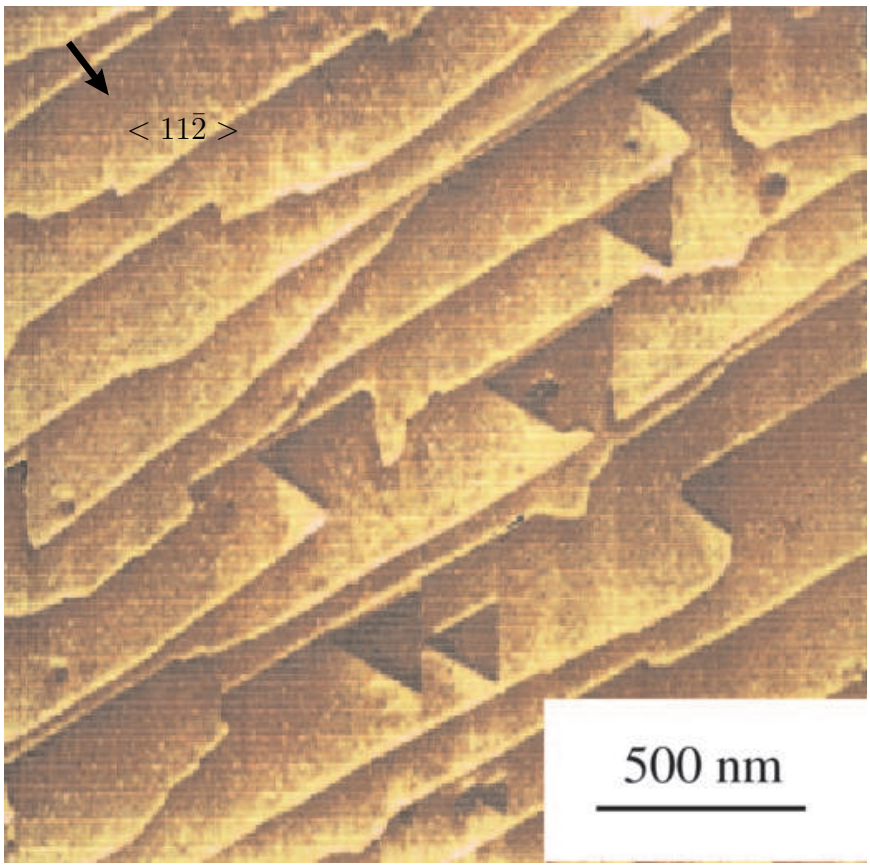


Figure 2.4. STM image of Si (111) surface after etching in NH_4F. The sample being etched has a miscut of 0.1° toward near < 11$\bar{2}$ >. It was etched in a 40% NH_4F aqueous solution for 15 minutes. The STM measurement was performed under UHV conditions.

2.4.3 Early theoretical models of preferential etching dynamics

2.4.3.1 Geometric-Kinematic Theories

Anisotropic etching is an old concept developed on the basis of early observations that a given etching solution would etch a crystal surface differently depending on the orientation of the crystal surface [45]. Intrinsically, because of the anisotropy of a crystal lattice, surfaces with certain orientations are etched more slowly than surfaces with other orientations. This is especially pronounced in silicon etching. For example, using a common silicon etchant, 50% KOH, the Si (111) plane etches about 600 times slower than the (110) plane [44].

In the geometric-kinematic theories, the surface is modeled as small elements of crystal planes. All surface elements having the same orientation share the same etch rate. This etch rate is called shift velocity. For purposes of simplification, assume that there are only two surface orientations: the vertical (face) plane and the horizontal plane (Figure 2.5). The elements in the face plane orientation have a shift velocity V_f, while those in the horizontal plane orientation have a shift velocity V_h. If V_f is larger than V_h, etch pits will be formed with the slope of the pit walls equal to V_f/V_h (Figure 2.5(a)). If $V_f \ll V_h$, as is the case when Si (111) is etched in NH_4F, the etching takes place in the form of layer-by-layer surface removal, which results in a microscopically smooth surface(Figure 2.5(b)).

The geometric-kinematic model is a very coarse model. While it can explain

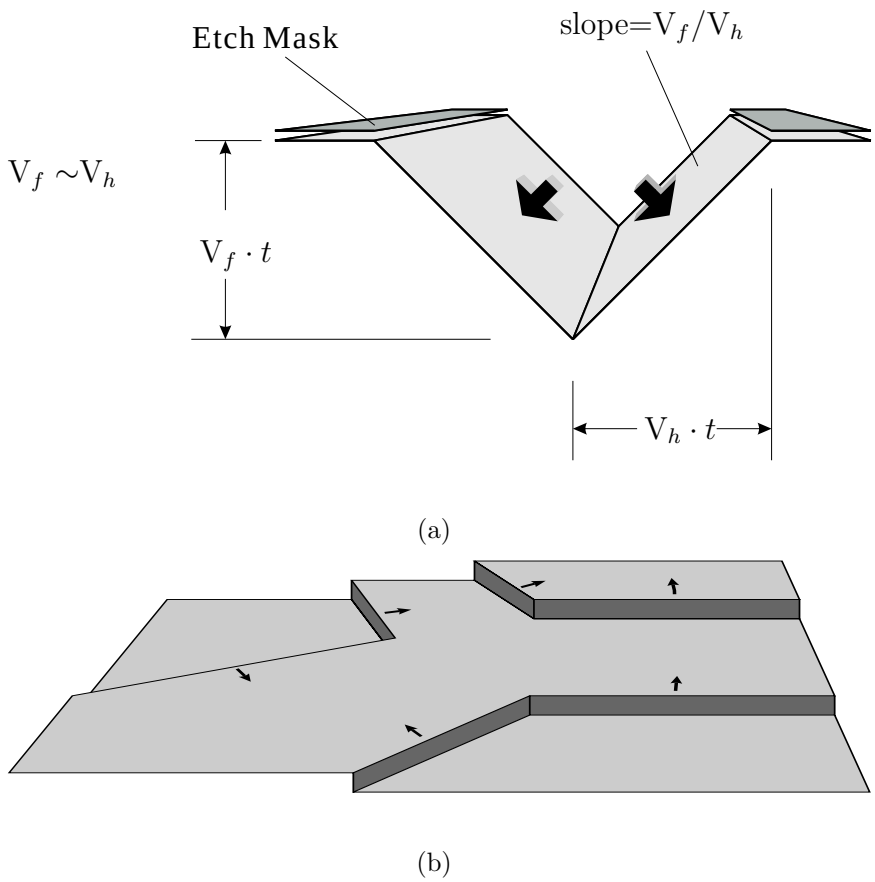


Figure 2.5. Illustration of preferential etching dynamics. (a) The generic situation of anisotropic etching. The slopes of the pit walls depend on the ratio of V_f to V_h. (b) For $V_f \ll V_h$, the etching reduces to layer-by-layer surface removal.

the chemical polishing effect of the preferential etching and the formation of the macro etch pits [46], it is difficult to use it to interpret the formation of macro steps. For the etching of Si(111) in NH_4F, where almost all the etching takes place as step flow, it is difficult to define the physical meaning of V_f. Thus it is rather limited in interpreting the surface morphology which develops in this case.

2.4.3.2 Molecular-kinetic theories

By far the most commonly invoked theory on silicon surface morphology development is the molecular-kinetic theories proposed by Frank [47, 45] and Cabrera [48]. For an excellent review, refer to Heimann's book [45]. This theory was recently revisited by Hines in an SEM study of Si etching in KOH [44].

In the molecular-kinetic model, etching is described as shift of surface steps. The flow rate of individual step L is assumed to a function of the local step density ρ.

$$L = f(\rho). \tag{2.6}$$

This model is very effective in describing the formation of ledges (figure 2.6). If $dL/d\rho < 0$ (the smaller the step density is, the faster a step propagates), an original uniform steps will transform into a "positive" bunching effect. If $dL/d\rho > 0$ (the larger the step density is, the faster a step propagates), an original uniform steps will transform into "negative" bunching effect.

Si(111) etched in NH_4F is a very slow etching. In our experiments, almost all the steps observed are single atomic steps. It's rare to observe the step bunching effect. Both Hines' study and our research on NH_4F etching of Si show $dL/d\rho = 0$. Thus the Frank-Cabrera model has limited use in explaining the surface morphology for Si(111) etched in NH_4F.

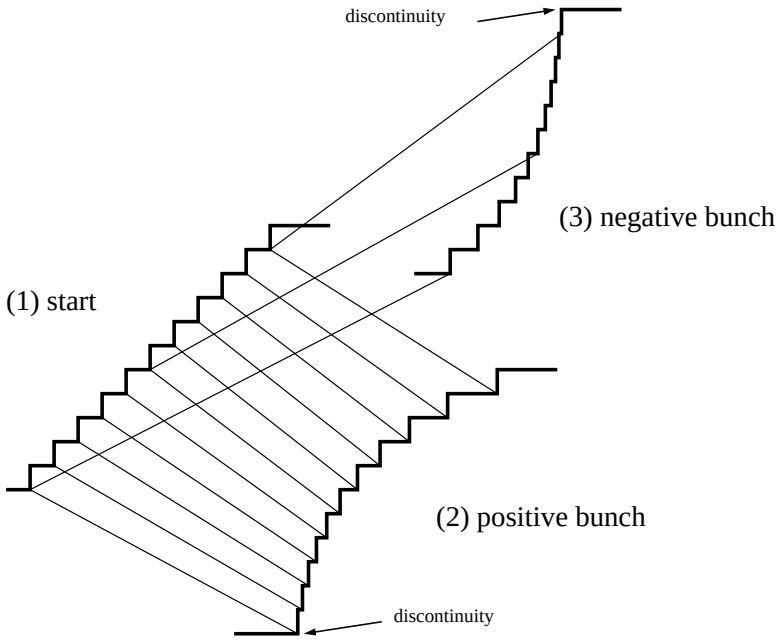


Figure 2.6. Illustration of the formation of ledges [45]. The original uniform step profile (1) will transform into (2) "positive" bunch if the steps in the region CD move slowly, (2) or "negative" bunch if the steps in the region CD move fast.

2.4.4 Step-flow model

The step-flow mechanism was first introduced by Hessel [21]. In this mechanism, two distinct surface activities are emphasized: the step flow and etch pit initiation. The pit initiation is a result of removing Si atoms from ideal (111) terraces. It leads to the nucleation of "holes" on the surfaces, which therefore increases the surface roughness. The step flow is a result of removing Si atoms from surface steps. The step-flow recovers the flat (111) surface terraces. These two activities form a competition in the surface morphology effect. This competition is affected by both the chemical environment and the surface miscut configuration. The result of this competition reveals various surface etch morphologies.

2.5 Surface morphology observed on samples with various wafer miscuts

On an ideally flat Si (111) surface with a miscut, the average terrace width W is determined by the wafer miscut angle, α:

$$W = h_s/tan(\alpha), \tag{2.7}$$

where h_s is the height of a single step (0.314 nm). Because of our interest in obtaining large, atomically flat terraces, we would like to study the surface morphology developed on surfaces with small miscut angles. However, our

experiments reveal that there is a limit on the maximum terrace width that can be obtained through the wet-chemistry process. For samples with small wafer-miscut angles, the morphology differs dramatically from that developed on samples with large miscut angles. The etching time required to reach a steady-state morphology was also found to be different for samples with different wafer-miscut angles.

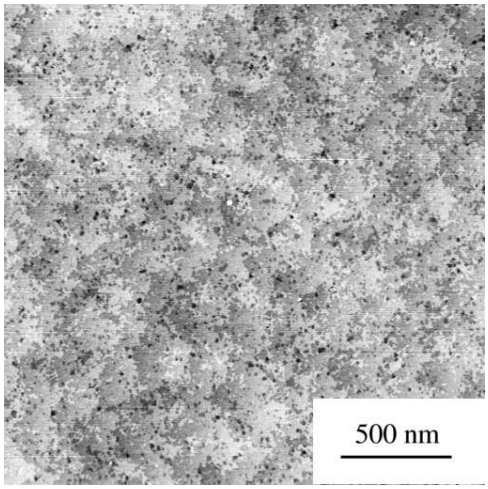
In the following experiments, the miscut angles of the surfaces are obtained from the average terrace width and equation 2.7. Both the wet chemical method and high temperature annealing method were used to obtain regular single steps and atomic terraces interlaced surface morphology to determine the average terrace width and miscut orientation.

Figure 2.7 shows a series of AFM images measured from samples with a wafer that has a miscut angle of 0.12° toward about 12° off $< 11\bar{2} >$. The miscut angle gives an average terrace width of 120 nm. All the samples were diced from a single wafer and went through the pre-cleaning procedures together, in one batch. Afterward, they were etched in the same 40% NH_4F solution. The samples were withdrawn from the etching solution sequentially, after 1 minute, 2 minutes, 4 minutes, and 6 minutes of etching, respectively. An AFM operated in air was used to observe the morphology of the samples within a few days after the etching.

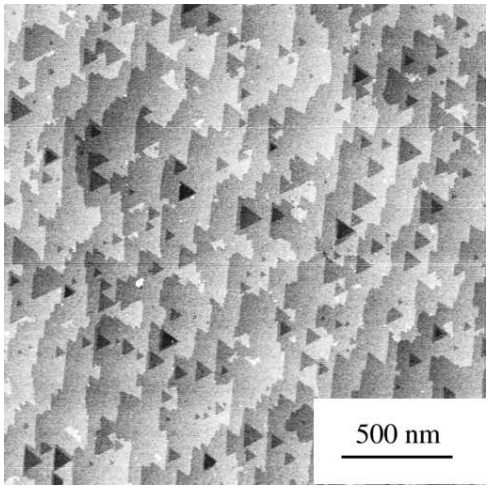
Just before the final etching, the silicon surfaces bore a thin layer of chemical oxide resulting from the RCA cleaning process. During the initial NH_4F etching,

this oxide layer was removed. After 1 minute of etching (Figure 2.7(a)), the oxide layer was fully removed. However, the surface was still atomically rough. There were numerous small etch pits all over the surface. The terrace structures were starting to show up but were very difficult to recognize. After 2 minutes of etching (Figure 2.7(b)), the pit density was greatly lowered and long-range terraces started to show up. The etch pits had grown larger, and long-range steps started to form as a result of pit collisions. After 4 minutes of etching (Figure 2.7(c)), the pits were even larger and most of them had merged into long-range steps. The steps were zigzagged as a result of the merging that took place. After 6 minutes of etching (Figure 2.7(d)), the surface become a periodic step-and-terrace structure, with no observable pits in the area being imaged. Extending the process to etching times longer than 6 minutes did not change the overall surface morphology, indicating that the etching of the surface had reached a steady state.

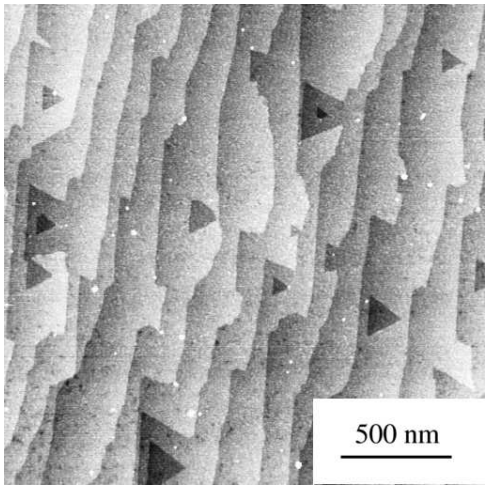
Experiments on samples with a much smaller miscut angle revealed quite different morphologies. Figure 2.8 shows a series of AFM images of samples from a wafer with a miscut angle of 0.02°. This miscut angle gives an average terrace width of about 1 μm, provided that the surface is perfectly flat. The samples were treated similarly as in the previous experiment, but for different etching times. As in the previous case, the surface was atomically rough immediately after removal of the oxide layer, but it was smoothed by growth of the etch pits once the etching had begun. Since in this case the natural terrace width was 1



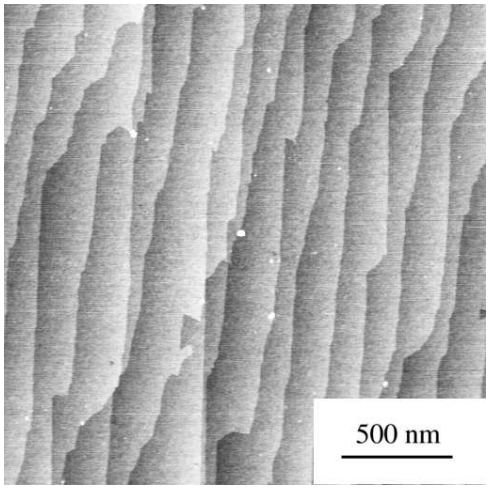
(a) 1 min.



(b) 2 min.



(c) 4 min.



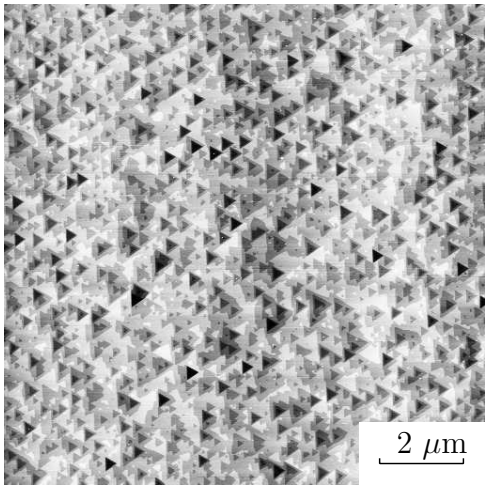
(d) 6 min.

Figure 2.7. Series of AFM images of samples with a miscut angle of 0.12° etched in a 40% NH_4F aqueous solution for (a) 1 minute, (b) 2 minutes, (c) 4 minutes, and (d) 6 minutes. All these samples were cut from a single wafer and etched in the same solution at the same time, but they were retrieved sequentially. The AFM measurements were performed by Joseph Fu of NIST.

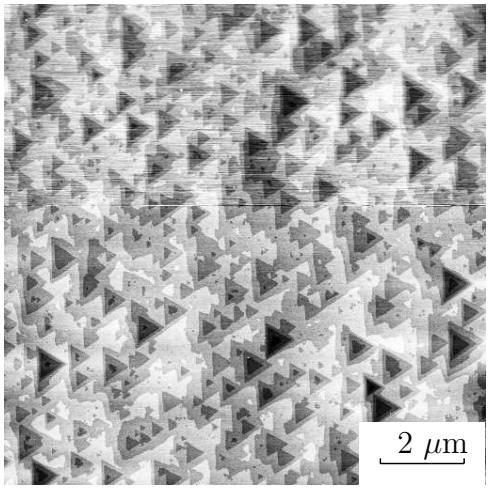
μm, collisions among triangular pits produced no long-range steps after 5 minutes, 15 minutes, 45 minutes, or even 3 hours of etching time. The surface morphology evolved as the average size of the triangular pits grew, and there was a concomitant reduction in pit density. After 45 minutes of etching, the surface morphology evolved into a steady state where the average pit size and pit density remained almost constant. Multiple pit-stacking patterns formed on the surface after an extended etching period. After 3 hours of etching, a multiple stacking pit with over 10 layers spread over an area of 4 μm was developed in the surface area we imaged (figure 2.8d). Because of the development of these stacking pits, the overall flatness of the surface was actually slightly reduced after the etching reached a steady state.

Figure 2.9 shows AFM images of three samples with different miscuts. All the samples were etched in NH_4F for 15 minutes. The samples with miscuts of 0.12° and 0.09° both exhibit long-range steps and terraces. The one with a miscut angle of 0.09° has more zigzagged steps, and a few triangular pits appear on the terraces. In the sample with the extremely small miscut angle, long-range terraces are not apparent and the pits dominate the surface.

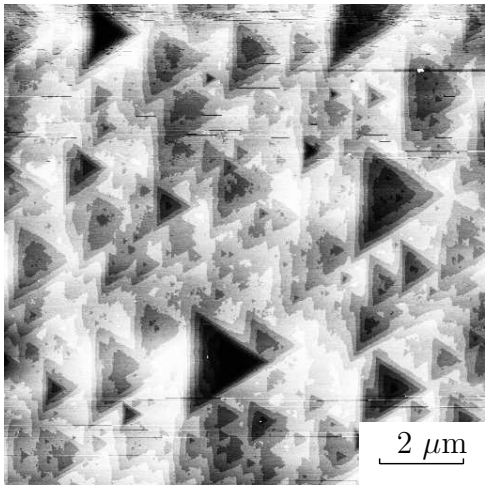
From our experiments, we found that the wafer miscut angle of the sample being etched had to be larger than about 0.1° in order to develop into a morphology with long-range steps and terraces. Moreover, to reach such a steady-state morphology, there was a certain threshold in terms of etching time. What we found is that the smaller the miscut angle of a sample, the longer the



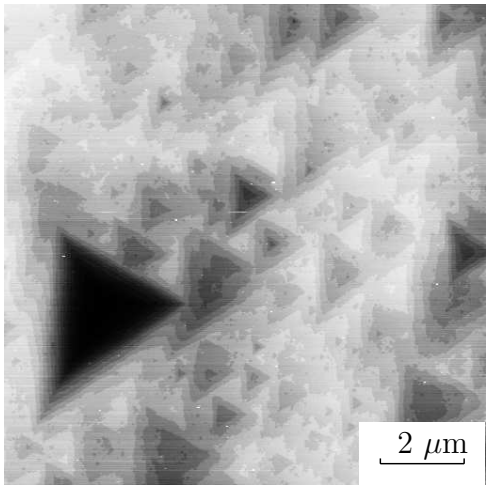
(a) 5 min.



(b) 15 min.



(c) 45 min.



(d) 3 hr.

Figure 2.8. Series of AFM images of samples with a miscut angle of 0.02° approximately facing $< \overline{1}\overline{1}2 >$ etched in 40% NH_4F aqueous solution for (a) 5 minutes, (b) 15 minutes, (c) 45 minutes, and (d) 3 hours. All these samples were cut from a single wafer and etched in the etching solution at the same time, but they were retrieved sequentially. The AFM measurements were performed by Joseph Fu of NIST.

time required for the initial etching to reach the steady state. For samples with extremely small miscut angles, the surface morphology reached a steady state after about 45 minutes of etching, but without the development of long-range steps and terraces. On such surfaces, the extra long etching time brought about a slight increase in the surface roughness on account of the formation of multiple stacking etch pits.

2.6 Discussion of the development of surface morphology

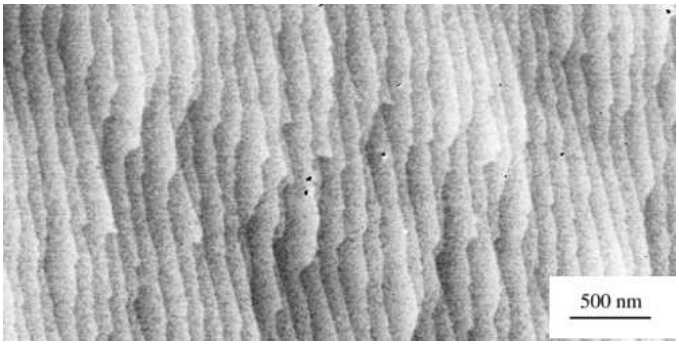
2.6.1 Step flow revisited

Before proceeding to a further discussion of the development of the surface morphology, it is beneficial to revisit the idealized step-flow model. An idealized steady-state Si (111) surface undergoing the etching process is illustrated in Figure 2.10. The surface exhibits a perfect staircase structure, where each terrace is an atomic plane and each step is a single atomic step with a height of 0.314 nm. The average terrace width W can be determined from the miscut angle:

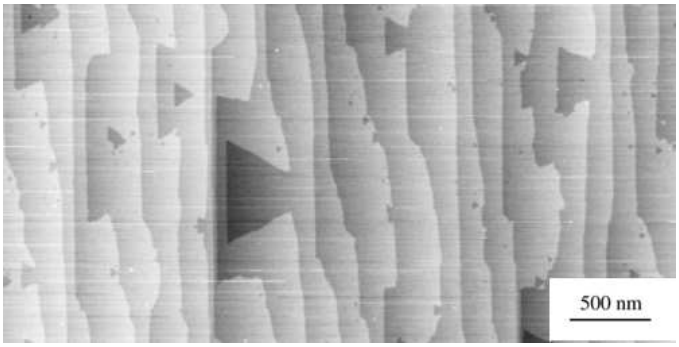
$$W = \frac{0.314}{tan(\theta)}\text{nm} \tag{2.8}$$

where θ is the miscut angle.

For an idealized step-flow etching, all the etching takes place at the step edges. From the morphology point of view, the etching takes place as a uniform



(a) 0.12°



(b) 0.09°

(c) 0.02°

Figure 2.9. AFM images produced from samples with different miscut angles. All the samples were etched in 40% NH_4F for 15 minutes. The AFM measurements were performed by Joseph Fu.

flow of all the steps in the same direction (namely, the direction of the miscut). Furthermore, the overall morphology does not change as the etching proceeds. The *critical etching time* is defined as the average time T_w required for each step to flow over a distance corresponding to the average terrace width W:

$$T_w = \frac{W}{L}, \tag{2.9}$$

where L is the average rate of step flow and T_w is the time required to etch away a single mono-layer (ML) of surface atoms. The vertical etch rate V can be calculated using the following equation:

$$V = 0.314/T_w = L * tan(miscut). \tag{2.10}$$

This equation can be used to deduce the step-flow rate from the measurement of the overall vertical etch rate.

As the steps flow at a uniform rate, the lifetime of certain terrace sites can be determined from the distances D of those sites from the previous advancing step edge:

$$t = D/L. \tag{2.11}$$

The lifetime of the terrace sites changes linearly across the terrace, and T_w is the maximum lifetime of terrace sites on such surfaces.

In the above discussion, the steps are limited to one dimension. For two dimensional steps, the orientation of the step also contributes to different step-flow rate. Steps facing $< 11\bar{2} >$ directions are packed with mono-hydrides,

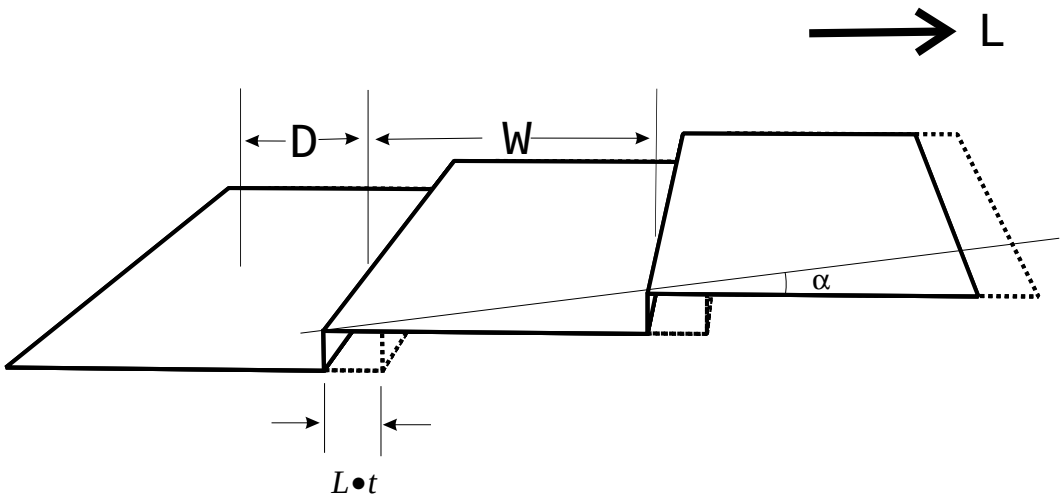


Figure 2.10. Illustration of an ideal step-flow etching

while steps facing $< \overline{1}\overline{1}2 >$ are packed with di-hydrides. The di-hydride silicon atoms are easier to be removed from the bulk crystal, thus steps facing $< \overline{1}\overline{1}2 >$ are expected to have a much larger step-flow rate. The direct determination of the step-flow rate was attempted by Ye [49] using electrochemical scanning tunneling microscopy (ECSTM). By continuously scanning the the surface during etching, they obtained a step-flow rate of 28 nm/min for steps facing $< 11\bar{2} >$ and 86 nm/min for steps facing $< \overline{1}\overline{1}2 >$.

If we were to take a further look into the etching on a surface step, we would be viewing the etching at the atomic level. A step at the atomic level consists of straight segments, together with kinks that separate the straight segments (figure 2.11). The kinks are being removed faster than other step atoms due to its exposed nature. The etching on a kink site doesn't eliminate the kink but

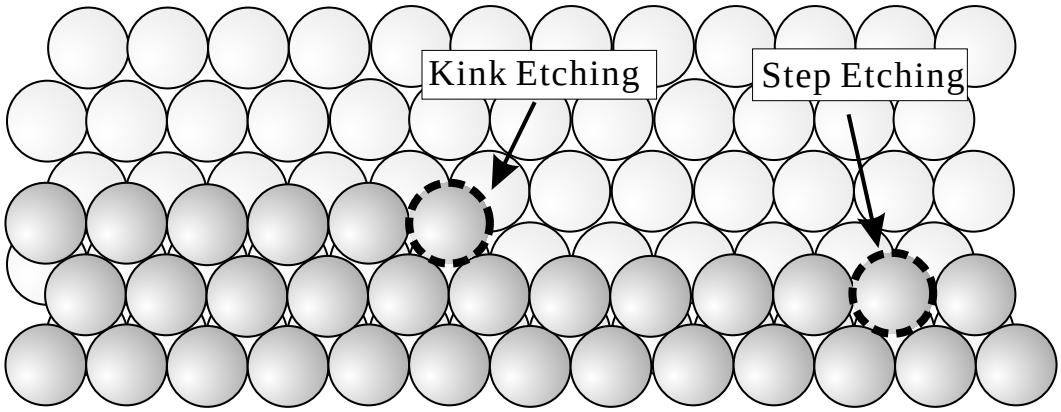


Figure 2.11. Kink sites are the corner sites on a non-straight single step. The population of kink sites indicates the roughness of the step. In a general etching situation, the kink atoms are etched away at a faster rate than those on a straight step. A fast kink etching lowers the population of the kink sites, thereby straightening the steps. However, the etching of step sites on a straight step creates more kink sites, thus increasing the roughness of the step.

propagates the kink position and over all it takes place as kink-flow etching. The relative density of kinks on a step contributes the overall step-flow rate. Thus, steps that are oriented slightly off the $< 11\bar{2} >$ directions are expected to have larger step-flow rates than steps that face exactly the $< 11\bar{2} >$ directions.

2.6.2 Pit initiation

Physical etching deviates from the above idealized mechanism as a result of pit initiation. Due to the finite etch rate of surface atoms in a flat terrace, the

etching randomly removes surface terrace atoms and leaves holes on otherwise flat surface planes. The holes creates new etch fronts and start growing as the etching proceeds and leads to observable etch pits. If we treat the etching of a given terrace site as a uniform, random process, then the probability of a certain terrace site being etched can be expressed as

$$P_t = 1 - exp(-t/t_0), \tag{2.12}$$

where t is the lifetime of the terrace site and t_0 is the average lifetime of all terrace sites if the random terrace etching is the only etching process that terminates the terrace site. t_0 is determined from the terrace site reactivity. In physical etching, terraces are mostly updated by the step-flow, thus the actual average lifetime is determined by step-flow rate and step distribution, which is much less than t_0. Given a certain surface area, suppose the total number of terrace sites is N, then for that certain surface area, the average number of pits initiated on this surface can be described as

$$N_p = N(1 - exp(-t/t_0)), \tag{2.13}$$

where t is the average lifetime of the terrace sites.

Equation 2.13 predicts that there will always be large amounts of pits initiated at the early stages of terrace lifetime. However, this conflicts with our experimental observations. Our experiments on Si(111) with large miscut angle such as 0.12° show that after long enough etch times, there are almost no pits observable anywhere on the wafer. The experiments indicate that there should be

some dormant time after a pit is initiated, during which the pit size is too small to be directly observed and grows very slowly.

A pit grows as a result of the step etching on its three borders. Although pit edges are believed to be the same in atomic composition as long-range steps, their flow rate is likely to be slower than that of long-range steps. This is illustrated by viewing the step etching as a kink flow. Similarly, the step etch rate in a kink-flow etching is determined by the kink density. For long-range steps, the kink propagates along the step until it collides with another kink, which results in kink annihilation. Often the overall step is facing in a direction that is not precisely the same as the low-index crystal direction. This results in creation of a series of kinks that face in the same direction and a lowering of the kink annihilation rate. For steps in a pit, however, each step is confined by the other two steps; thus the kinks are quickly annihilated once they propagate to the corner, resulting in a lower kink density in pit steps than in normal steps. It is also reasonable to postulate that the smaller the pit is, the lower the kink density on its steps will be—and the more slowly that pit will grow.

Figure 2.12(a) illustrates this non-linear pits growth curve. As a result of this curve, if we plot the pit size against the pits life time (or average pit size against the average terrace life time), we will have a curve as in figure 2.12(b). It takes a long initial time, T_p, for a single site etch pit to grow to the size at which it starts to affect the overall morphology (namely, a few nanometers). If we assume a

constant step-flow rate L, we can define a critical pitting distance D_p as

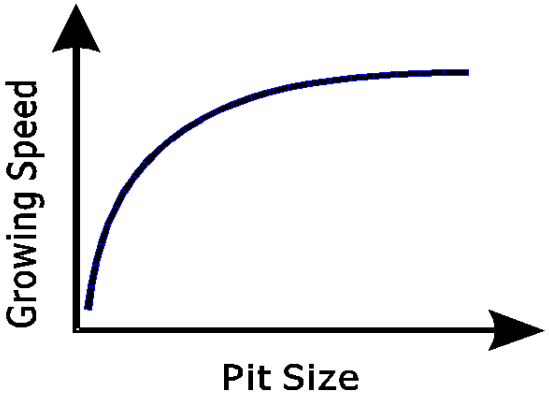
$$D_p = L \cdot T_p. \tag{2.14}$$

On average, a pit will become large enough to be observable only when its distance from the previous step edge is larger than D_p. Since the rate of growth of the pits increases as the pits get larger, it is reasonable to further postulate that pits which are formed earlier will engulf pits that are formed later, since the ratio of the sizes of a pair of pits tends to increase as the difference in their initiation times gets larger. For a surface etched under the ideal step-flow mechanism, we propose that the pits that contribute most to the overall morphology are those pits that are formed when their distance from the site to the previous step is on the order of D_p.

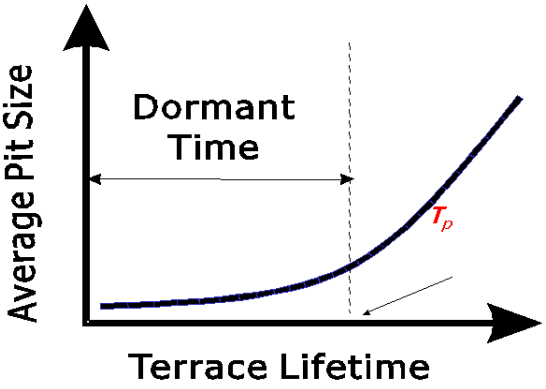
D_p and equation 2.13 contribute to the pitting effect differently. The terrace site etch probability, P_t in equation 2.13, is directly determined from the finite etch rate of a terrace site or alternatively the reactivity of the terrace site. This determines the density of pits at early terrace lifetimes. D_p is derived from the pit growth curve, it determines the dormant time of a pit after it is initialized.

2.6.3 Etching on surfaces with a large miscut

Figure 2.13 illustrates etching on surfaces with large miscut angles. This is the case whenever $D_p \gg W$, that is, when the critical pitting distance is much greater than the average terrace width. The age of terrace sites is kept short so



(a)



(b)

Figure 2.12. Illustrations of the non-linear pit growth curve. (a) The pit growing speed plotted against pit size. (b) The pit size plotted against terrace lifetime.

that at most an insignificant number of pits will be initiated during that time. For the occasional pit that is initiated, its size is kept very small because of the short time available for it to grow. The merging of small-sized pits and straight steps creates protruded corners; however, the size of such corners is very small, on the same order as the size of the kinks that are generated during the step flow. Thus, these pits generally do not affect the straightness of the long-range steps.

The case of surfaces with large miscuts is the one that comes closest to the ideal step-flow model. Pits on these surfaces are difficult to observe, on account of their small size. Due to its high step density, the over-all surface is etched at a high rate. Equation 2.10 gives the relationship between the over-all etch rate and average terrace width.

2.6.4 Etching on surfaces with a medium miscut

Figure 2.14 illustrates etching on surfaces with medium miscut angles, where the average terrace width is on the order of D_p. The maximum terrace lifetime on these surfaces is longer and allows for growth of etch pits to an observable size. As the pits merge into the advancing steps, one segment of the original straight step takes a jump ahead, leaving two corners on the steps. The corners provide a source of kink generation and enable the step to be etched even faster. Other things being equal, the step would become straight again as a result of the fast corner etching. However, new pits are constantly being formed and merging with

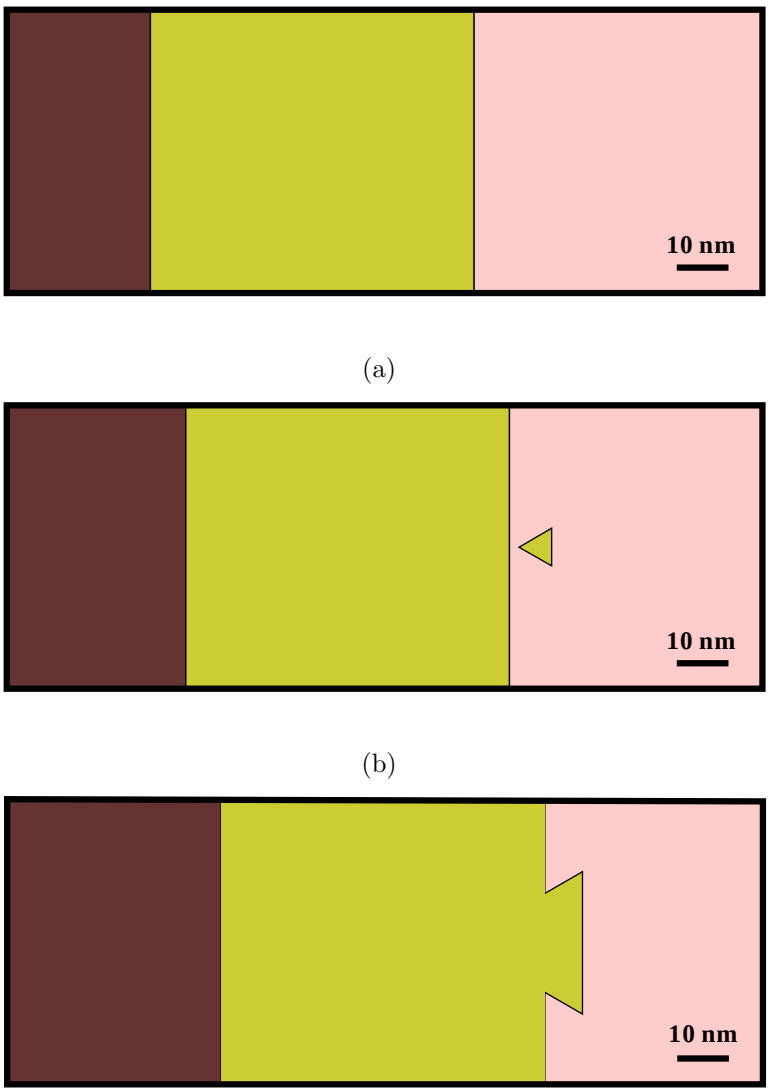


Figure 2.13. On surfaces with large miscut angles, the average terrace width is small and the average terrace lifetime is kept short on account of the step flow. On such surfaces, almost all the pits that are initiated are limited at very small sizes. The mergence of these pits to the advancing steps creates negligible effects on the overall morphology. Steps and terraces on these surfaces are straight and long range, pits are difficult to observe.

the steps, which causes the steps to become zigzagged. At higher pit initiation rates, the pits are formed at higher density, and the resulting merging with high-density pits can render the steps fragmented. Overall, the steps flow at a higher rate because of the pit merging, while long-range terraces are still being formed with scattered pits.

At the start of the etching process, numerous etch pits are initiated because of the initial roughness or surface defects. These pits grow and merge into larger pits and step edges. As the result of the merging, the initially created steps tend to become rather zigzagged. On a zigzagged step edge, the point sites or kink sites are more vulnerable to etchant attacks, the outcome being that the steps get straighter and straighter. For surfaces with larger miscut angles, the new pits that are created are limited in size and number, while the steps get straighter and straighter until they reach a steady state. For surfaces with smaller miscut angles, new pits are constantly being nucleated and merged into steps, so the steps may never get straighter. It thus seems reasonable to postulate that the magnitude of the zigzags of the step edges in the surface morphology in the case of steady-state etching is related to the average terrace width.

2.6.5 Etching on surfaces with a small miscut

Figure 2.15 illustrates etching on surfaces with very small miscut angles. On surfaces of this kind, $D_p \ll W$, and the average terrace width is large enough to

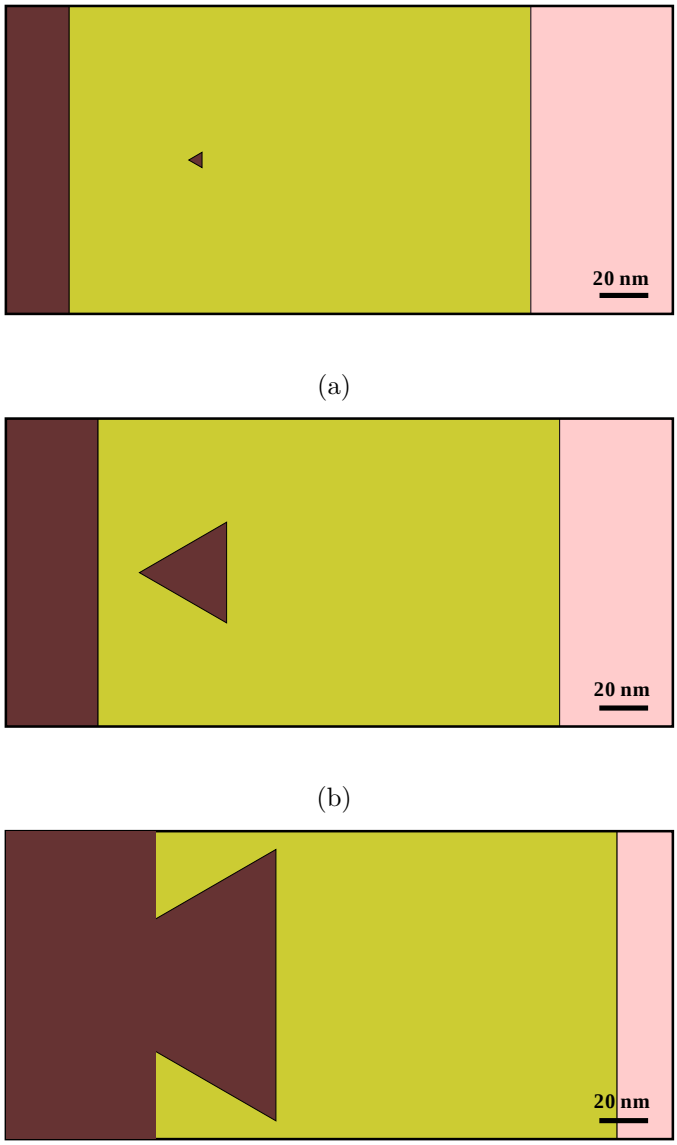


Figure 2.14. On surfaces with medium miscut angles, a significant number of pits are nucleated before the terrace is removed by the step flow, and these pits are capable of growing to an observable size. On such surfaces, the long-range steps and terraces are usually maintained; however, the constant merging of steps and pits causes the steps to become zigzagged. The magnitude of the zigzagging is determined in part by the average terrace width.

have new pits growing inside it. This is how the multiple stacking pits are initiated. As a random process, the newly generated pits may not necessarily have the same center as the parent pit. However, the initial collision of these new pits results in a pit that is approximately in the center. The new center pit expands along with its parent pit and another new layer of pits will be generated inside around the center. The merging of pits and steps creates long segments of zigzagged steps or even multilevel zigzagged steps. As a result, the original idealized step orientation loses its dominance and the long-range steps disappear. As indicated by Hines [50], there is a dynamic repulsion effect which results in equalized step-step spacings. In the case of very small miscut angles, that same effect contributes to alignment of step segments along the edges of multiple stacking pits, resulting in the formation of a layer of semi-closed pits. The outer pits protect the inner pits from collisions with other steps or large pits, and so the inner ones are capable of growing larger. The overall size of a stacking-pit pattern may slowly increase as the etching proceeds.

2.6.6 Effects of miscut orientation

In the previous discussion, a miscut toward the $< 11\bar{2} >$ direction was assumed. In experiments, the miscut orientation often deviates slightly from the ideal $< 11\bar{2} >$ direction. As proposed in section 2.6.1, for steps that face slightly off $< 11\bar{2} >$, the average rate of step flow increases as more kink sites are formed.

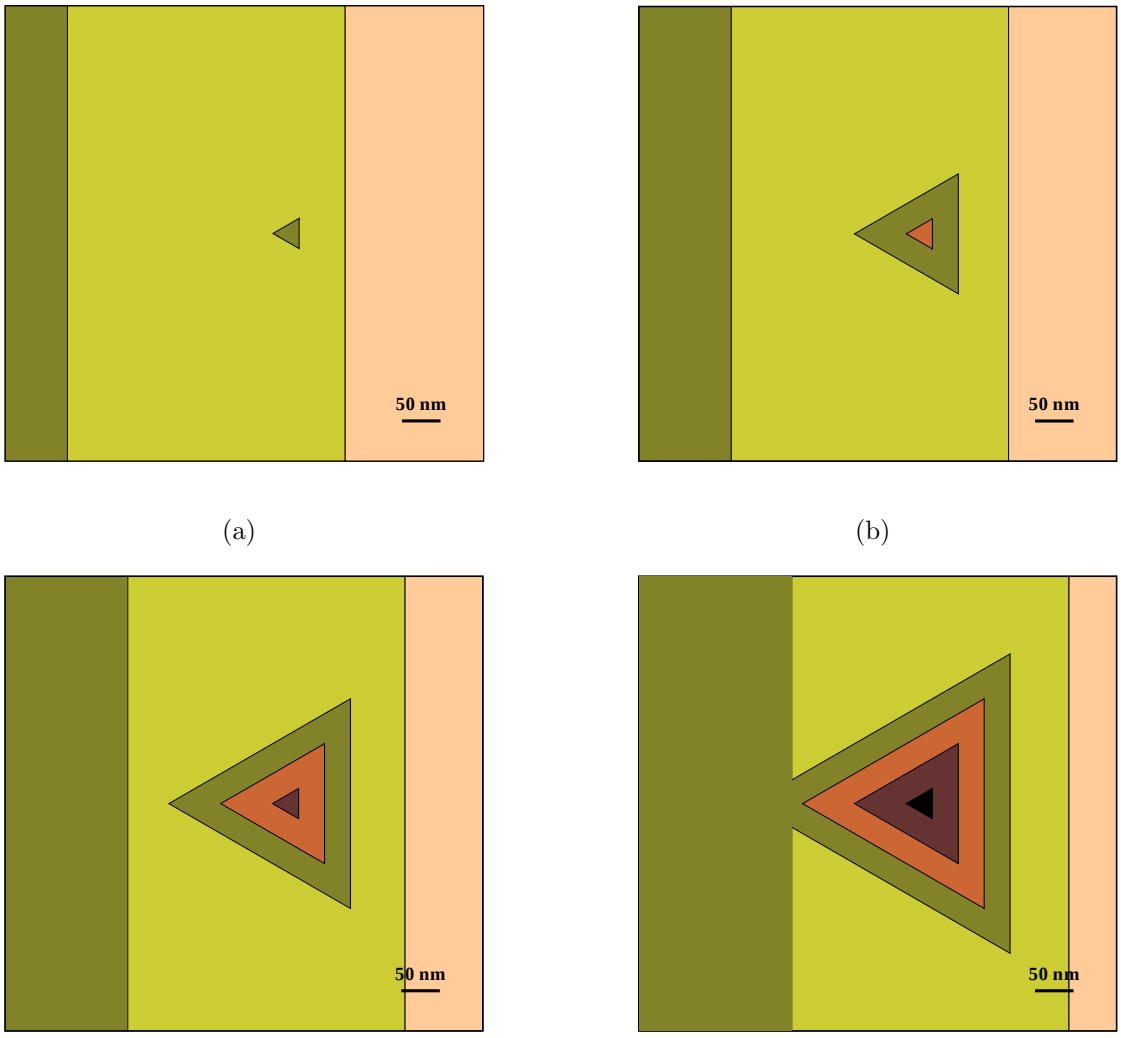


Figure 2.15. On surfaces with very small miscut angles, the pits can attain a very large size and new pits can be initiated inside older pits.

The increase in L results in a smaller value of T_w (equation 2.9) and a larger value of D_p (equation 2.14), consequently, an increase in the ratio D_p/T_w. As for the final morphology of the misoriented surfaces, although the steps suffer from regular zigzagging on account of the misorientation, they suffer less from the merging of the pits than they would in the case of exact orientation toward $< 11\bar{2} >$. In our results that are shown in Figure 2.9, the sample with a miscut of 0.12° is slightly misoriented, which is evidenced by the regular kinks on the steps. In comparing the morphology of that sample to the one with a miscut of 0.09°, we find that the latter exhibits much more in the way of pitting, though there is only a small difference in average terrace width for the two surfaces (150 nm versus 200 nm). This effect can be partially attributed to the increase in the step-flow rate.

2.6.7 Effects of variation in temperature

As is common to most chemical reactions, increasing the temperature will increase the etch rates of all the sites, while lowering the reaction temperature will decrease the etch rates. In general, the reaction rate follows the Arrhenius equation:

$$k = Ae^{-\frac{Ea}{RT}}, \tag{2.15}$$

where Ea is the apparent activation energy, which is the energy barrier height between the two states of the reaction. Because of the different structural

configurations that a sample can assume, different etch sites can have different energy barriers. We can express the rates of etching of terrace sites and step sites as in the following equations,

$$k_{terr} = A_{terr} e^{-\frac{E_{a(terr)}}{RT}}, \tag{2.16}$$

$$k_{step} = A_{step} e^{-\frac{E_{a(step)}}{RT}}, \tag{2.17}$$

$$k_{kink} = A_{kink} e^{-\frac{E_{a(kink)}}{RT}}. \tag{2.18}$$

For the first effect, the increase in temperature will increase the etch rate for terrace sites, thus increasing the density of pits that get initialized. (Equation 2.13). For the second effect, consider the the ratio

$$\frac{k_{kink}}{k_{step}} = \frac{A_{kink}}{A_{step}} e^{-\frac{E_{a(kink)} - E_{a(step)}}{RT}}, \tag{2.19}$$

We propose that $E_{a(kink)} < E_{a(step)}$ as in figure 2.16, and hence an increase in temperature will lead to an decrease in the ratio of the etch rates given above. As we discussed previously, the large difference between the etch rates of kink sites and step sites contributes to the different step-flow rates of long range steps and confined steps that enclose the pits, and thus contributes to the dormant time period. So the decrease in the ratio of activation energies, E_{kink}/E_{step}, contributes to this effect thereby resulting in a smaller value of D_p.

Figure 2.17 shows the surface morphology of samples from the same wafer as the one in Figure 2.7, but the etching was performed in a hot NH_4F solution. It took only one minute to reach its steady-state morphology, indicating an

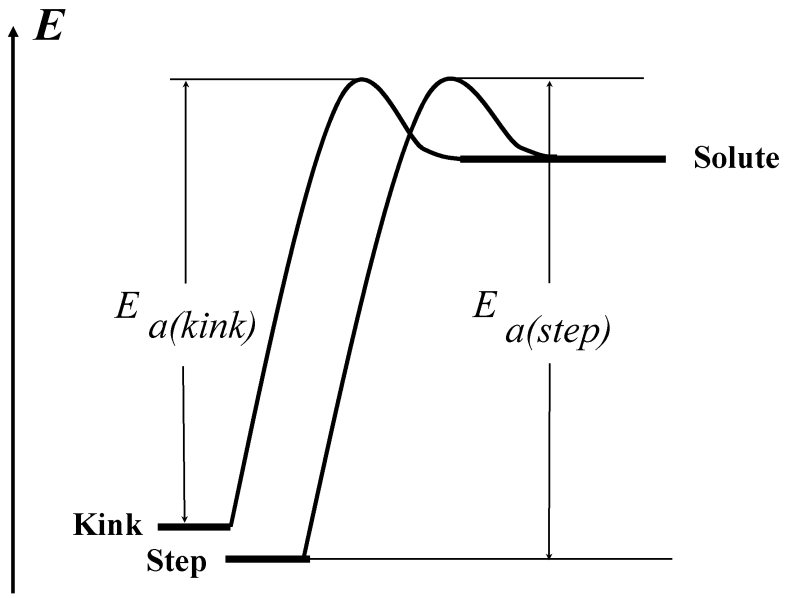


Figure 2.16. Illustration of activation energy of the etch reaction.

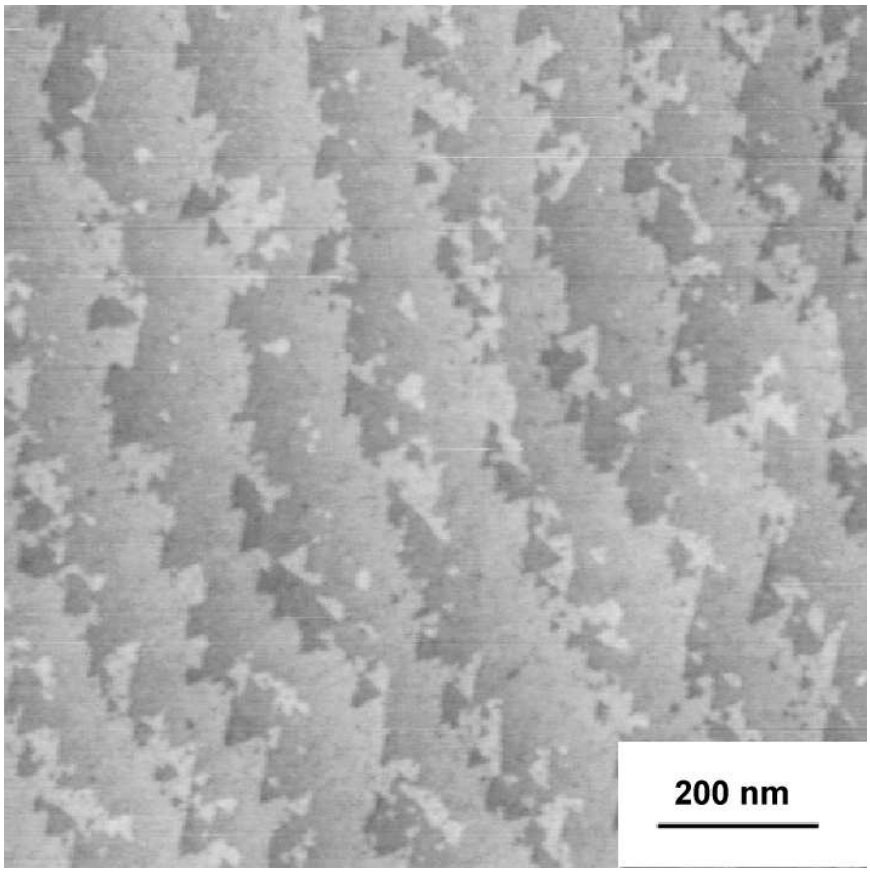


Figure 2.17. AFM image of sample with a 0.12° miscut angle etched in a hot NH_4F solution for 1 minute. During the etching, the temperature was not accurately measured. It was estimated to be $\approx 70°C$. The AFM measurement is performed by Joseph Fu of NIST

increased step-flow rate and a reduction in the critical wafer-etching time. The steps, however, are fragmented, even though on a large scale they appear to be straight. This is the result of constant merging of high-density pits which stemmed from a decrease in the value of D_p/W.

2.6.8 Conclusion

The surface morphology of a sample after wet-chemical processing is dependent on the etching time, the wafer miscut angle, and the wafer miscut orientation, as well as the etching conditions. Once the etching has been in progress for a long enough time, the surface develops into a more or less steady-state morphology. This final morphology is largely determined by the ratio D_p/W, where D_p is the critical pitting distance and W is the average terrace width. For $D_p/W \gg 1$, the steady-state morphology is dominated by long-range steps and terraces, with etch pitting highly suppressed. For $D_p/W \ll 1$, the morphology does not have long-range steps and terraces, and multiple stacked pits are randomly distributed across the surface.

2.7 STM imaging of the H–Si (111) surfaces

After a proper NH_4F etch, the surface is atomically flat. Higashi [34] have carried out imaging of surfaces using STM with atomic resolution; they have demonstrated that the surfaces are atomically ordered. Long-range atomically resolved STM measurements on such surfaces have recently been reported [11]. However, atomically resolved STM images on wet-chemically-prepared surfaces have rarely been reported, which may be an indication that such surfaces are extremely difficult to image via STM. This could be attributable to the hydrogen termination, since with hydrogen termination there are no surface states above

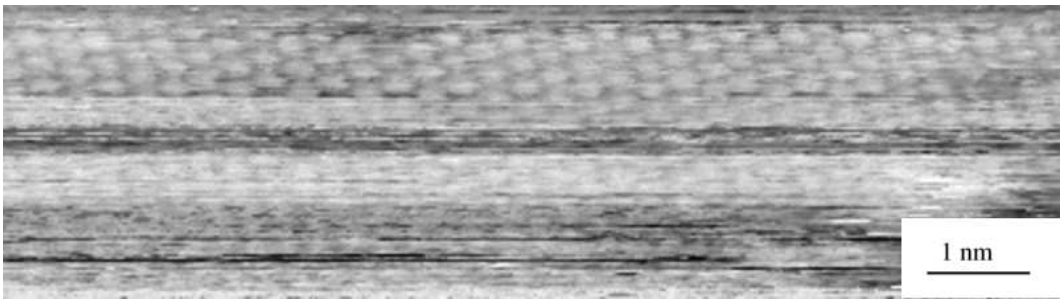


Figure 2.18. Atomically resolved STM image of hydrogen-terminated Si (111).

the Fermi level, thus rendering electron transfer through the tunneling barrier very inefficient. In our experiments, atomically resolved STM images were obtained (Figure 2.18), which demonstrated that the surfaces were atomically ordered. However, the tunneling condition was found to be unstable and difficult to maintain.

Chapter 3

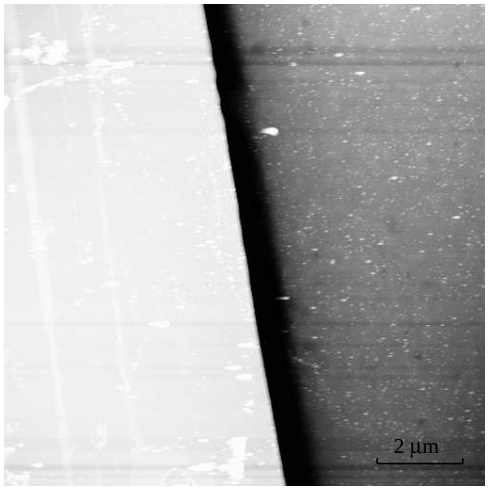
Kinetic Monte-Carlo Simulation

3.1 Introduction

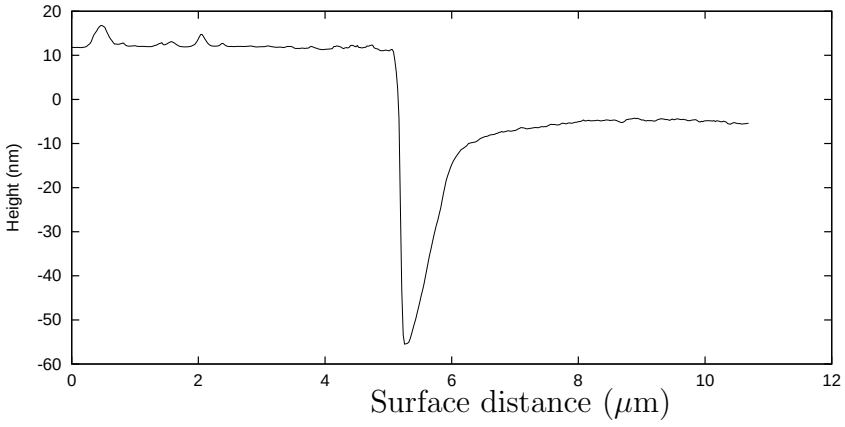
In the previous chapter, we presented a qualitative discussion of the development of Si (111) surface morphology during aqueous NH_4F etching using preferential etching dynamics. Three key components in that discussion were the step-flow rate, the pit growth rate, and the pit initiation rate. With these rates defined, it is possible to simulate the surface morphology development more precisely and gain further understanding of the process. To determine these etch rates experimentally, however, is very difficult due to the limit in the choice of silicon wafers and experimental techniques. Hines [24] introduced Monte-Carlo kinetic simulation which directly simulates the etching at the atomic level. With appropriate site specific etch rates, this simulation can be used to estimate the above mentioned quantities. In the following, I will briefly review the experimental methods and discuss our implementation of the kinetic Monte-Carlo simulation.

An intuitive way to experimentally determine the overall etch rate on a surface is by masking and directly measuring the surface height difference after a limited etching time. The etch rate obtained with this method is the apparent vertical etch rate. With the assumption of a uniform step-flow mechanism and ignoring the effects of etch-pit formation, equation 2.10 can be used to determine the average step-flow rate. As a preliminary test, we performed an experiment that used wax film as an etch mask. After 20 minutes of etching in NH_4F, the surface was cleaned and measured using an AFM under ambient environmental conditions. The result is shown in Figure 3.1. From the image, the vertical etch rate was estimated to be about 1 nm/min. Assuming an average terrace width of 120 nm (as calculated from 0.12° miscut angle) and a homogeneous step flow, each step was found to flow at a rate of about 380 nm/min. However, a trench as wide as 1 μm was created at the edge. Due to the limitation of AFM imaging, the exact profile of this trench could not be determined; and its formation mechanism is still under investigation. As a direct effect of this trench, the right side of the surfaces near this trench are expected to have a much higher step density than that resulted from 0.12° miscut angle. Thus, the etch rates obtained with this method are likely to be much higher than those etch rates should be on flat surfaces.

Ouyang [51] studied silicon etching in NH_4F by using a method similar to ours. In their experiments, a photomask was used to mask the surface with 2 μm $\times$ 2 μm squares, and the etch rates for a variety of surfaces—etched under



(a)



(b)

Figure 3.1. (a) AFM image of a Si (111) sample etched in 40 % NH_4F for 20 minutes. The left side of the sample surface was masked with a wax film before etching. The wax mask was cleaned after the etching and AFM was used to measure the etch depth. (b) Plot of cross-sectional profile of the same image. The orientation of the wax edge was not recorded in this experiments. The AFM measurement was performed by Joseph Fu of NIST.

a variety of etching conditions—were studied. For Si(111) etched in oxygen free NH_4F, they obtained a vertical etch rate of 0.08±0.06 nm/min. However, due to the narrow edge-to-edge spacings (2μm), the chemical environment in the holes on the surfaces may be different from that on a flat surfaces, thus the etch rates obtained by Ouyang should not be directly relied on either.

Ye [49] studied the etching of Si (111) surfaces in NH_4F *in situ* with electrochemical scanning tunneling microscopy (ECSTM). In their experiments, a surface area of 500 nm × 500 nm was dynamically measured by STM every 30 seconds over a period of two minutes. The positions of several steps were tracked and the step-flow rates for each were derived. They obtained step-flow rates of 28 nm/min for the step facing $< 11\bar{2} >$, and 86 nm/min for the step facing $< \overline{11}2 >$. These results were obtained from a single observation, thus, the uncertainties were not determined.

Instead of directly measuring the etch rates, it is also possible to derive these etch rates using simulations. With a satisfactory theoretical model, computer simulations can be used to simulate the surface morphology evolution during etching. The etch rates are generally a set of parameters for the simulation. By comparing the morphologies from the simulations and experiments, an optimum set of parameters may be obtained to achieve the best agreement. With an ideal theoretical model, the optimum parameters can give a good estimation of the physical etch rates.

There are generally two approaches to modeling the surface morphologies.

The first approach is based on the observables of the surface morphology — mono-atomic steps. The evolution of the surface morphology is determined solely by surface step migration. This approach fundamentally is the Cabrera-Franck model [47] and, as an example, it was applied to model the current-induced step bending effect on Si(111) [52]. However, in the study of surface morphology on Si(111) etched in NH_4F, as we discussed in previous chapter, in addition to long range step migration, the initiation of etch pits also significantly contributes to the final morphology, which makes it very difficult to directly apply the Cabrera-Franck model.

The second approach is the Monte-Carlo simulation based on the solid-on-solid (SOS) model [53, 24]. The SOS model assumes that the crystal lattice is rigid and that the atoms stay at the lattice points until they are etched away. The algorithm for simulation based on this model is simpler than the former approach. Hines successfully demonstrated the effectiveness in using kinetic Monte-Carlo simulations to study the surface morphology on Si(111) etched in NH_4F [24]. By comparing the resulting surface morphologies of their simulations to their experiments, they obtained an optimum set of site-specific etch rates. Hines further used this simulation to interpret the effects of step–step interactions [50] and the formation of hillocks [36], as well as to determine the pH level of the etching solution [54] and to investigate the effects of adding isopropanol [55] or dissolved oxygen [40] to the etching solution.

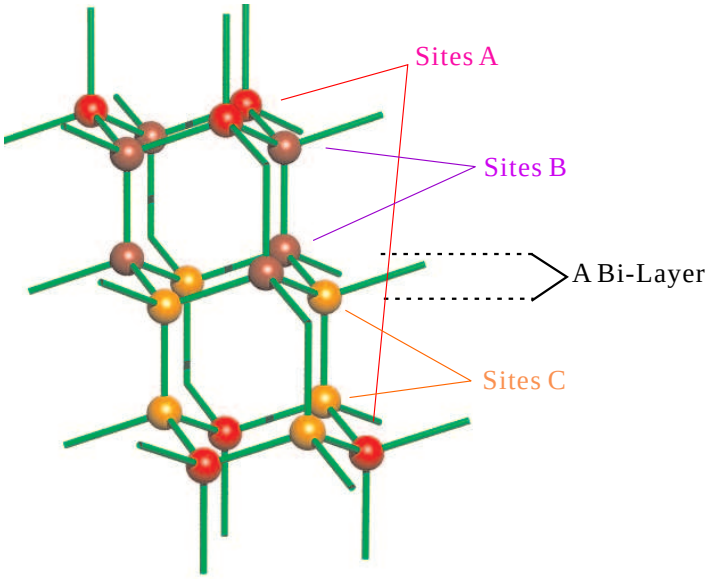
Although the set of site-specific etch rates obtained by Hines are not directly

linked to the surface step dynamics, they do provide a way to study the step dynamics by using this Monte-Carlo simulation for "theoretical experiments", and lead to our further understanding of the evolution of the surface morphology. In this chapter, we will use kinetic Monte-Carlo simulations, based on their published optimum parameters [36, 55] to study some of the key aspects of surface morphology development during NH_4F etching of Si (111).

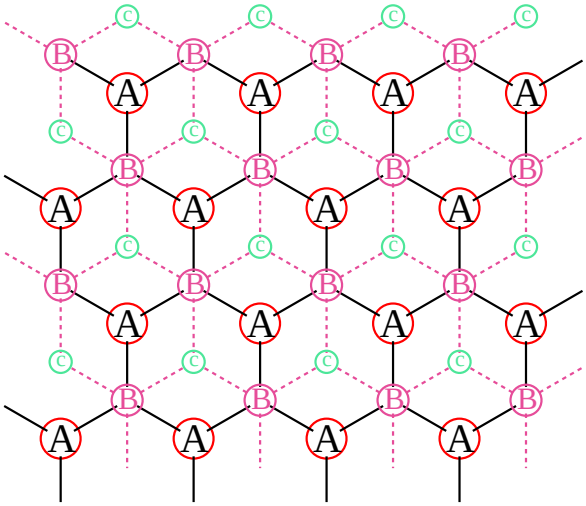
3.2 Preferential etching at the atomic level

The silicon lattice has a diamond lattice structure. The Si (111) surface lattice is essentially a network of six-membered rings in the "chair" configuration (Figure 3.2a). This network forms a "bilayer", since the lower surface atoms are depressed by 0.08 nm toward the bulk. The bilayers are connected by vertical Si–Si bonds, with a spacing between bilayers of 0.314 nm. From a top view, the sites on a given level of the Si (111) surface form a network of centered hexagons (sites "A"). The neighboring level is a network with the same structure but displaced in the $< 11\bar{2} >$ direction (sites "B" and "C"). The overall Si (111) lattice is an AB–BC–CA–AB–BC–... structure (Figure 3.2b).

On a Si (111) surface, the following atom sites can be treated as different species (Figure 3.3), provided that the surface is always hydrogen terminated during etching: terrace mono-hydride, step mono-hydride, vertical di-hydride, horizontal di-hydride, terrace tri-hydride, step tri-hydride, kink, and point.

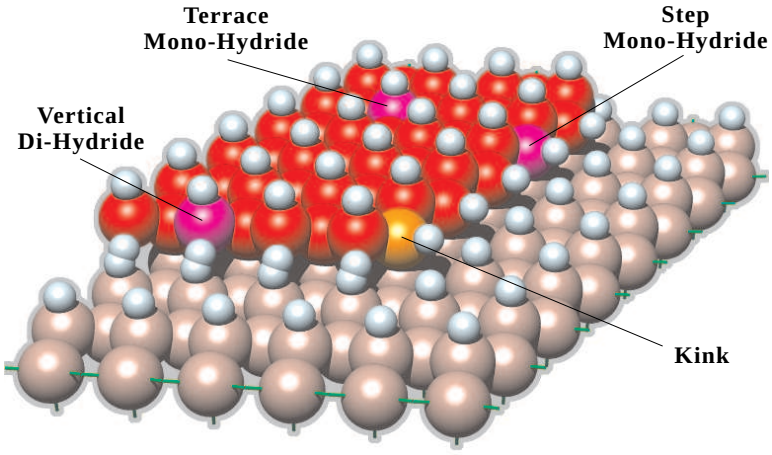


(a)

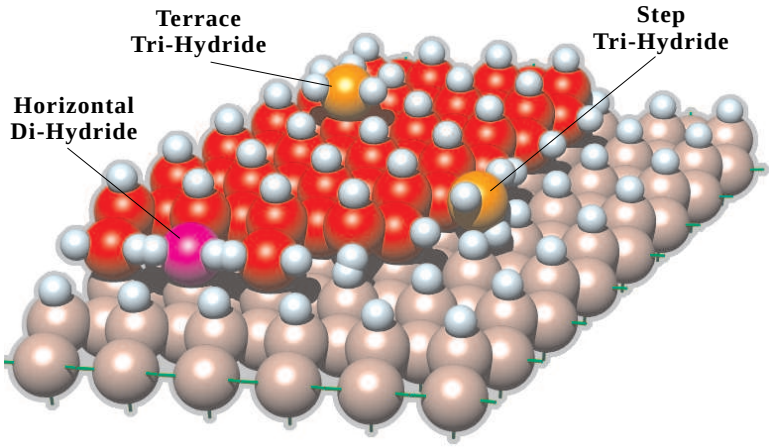


(b)

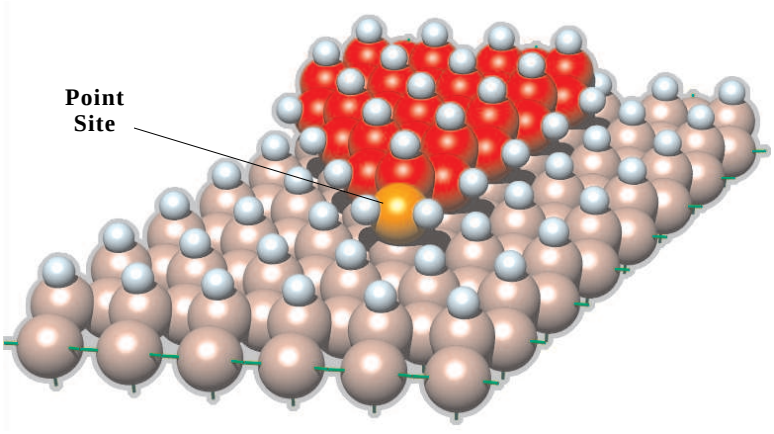
Figure 3.2. Illustration of the Si (111) surface lattice. (a) A ball-and-stick model showing the Si (111) lattice as a network of "chair" configuration. (b) A top view of Si (111) lattice sites. The surface lattice essentially consists of three set of hexagonal lattice: A, B and C. Each bi-layer is a combination of two set of lattice sites. The crystal lattice is an AB–BC–CA–... structure.



(a)



(b)



(c)

Figure 3.3. Atomic structure of a hydrogen-terminated Si (111) surface.

The major sites on flat terraces are terrace mono-hydride, the major sites on steps facing in the $< 11\bar{2} >$ direction are step mono-hydride, and the major sites on steps facing in the $< \overline{11}2 >$ direction are vertical di-hydride. These sites have all been identified via IR spectroscopy [56]. The stability of these three classes of sites is an indication of the stability of terraces, $< 11\bar{2} >$-facing steps, and $< \overline{11}2 >$-facing steps, respectively.

From a structural point of view, the mono-hydrides are the most stable sites, as they are not subject to strains of any kind; furthermore, three Si–Si bonds must be broken to etch away a single mono-hydride atom. The terrace mono-hydride is more stable than a step mono-hydride, since the former bonds to three bulk atoms, while the latter bonds to two surface atoms. A similar argument can be made for the stability of terrace mono-hydrides at the edges, which are bonded to two step mono-hydrides. To simplify the computations, we made no distinction between the edge terrace sites and other terrace sites.

The vertical di-hydride is bonded to only two bulk atoms, and infrared studies [56] indicate that the structure is highly strained (in particular, it is rotated by $\approx 28^\circ$ from its bulk-terminated position) because of the steric hindrance caused by the presence of the mono-hydride on the lower terrace. The vertical di-hydride is thus expected to be more reactive.

It is also possible to terminate a step facing $< \overline{11}2 >$ with horizontal di-hydride, and on terraces and steps facing $< 11\bar{2} >$, tri-hydride may exist. The horizontal di-hydride, terrace tri-hydride, and step tri-hydride were treated as

transient species because of their apparent high reactivity. Their finite etch rates do not appear to change the overall morphology to any appreciable extent. To simplify the computations, therefore, their etch rates were simply treated as being infinite.

A kink site is the intermediate site that separates a step facing $< 11\bar{2} >$ and a step facing $< \overline{11}2 >$. A point site is the corner site resulted from two joint steps facing $< 11\bar{2} >$. Kink sites and point sites are expected to be etched differently than the step mono-hydride and the horizontal di-hydride, because the former structures apparently suffer from different types of strains and steric hindrances than the latter. More importantly, they are the key sites for the development of certain forms of surface morphology. The density of kink sites reflects the straightness of a step, and fast kink etching produces straight step segments [24]. The point sites play a key role in the formation of hillocks [36], which is a type of feature that arises from step–pit collisions and develops on steps that face in the $< \overline{11}2 >$ direction. Thus, the relative etch rates of these sites have to be treated properly to yield these particular morphological features.

3.3 Computational algorithm

In principle, our simulation program is similar to that of Hines [24]. In essence, the exact Si (111) surface lattice is represented, and the atoms are randomly removed according to a set of predefined etch rates. However, the program we

employed was developed independently and some aspects of the algorithm are different from Hines'. For example, it appears that Hines' software was not able to track the actual etching time during simulations. This difficulty does not exist in our software, and enables us to do time resolved comparisons between our simulations and experiments. Some key algorithms are discussed in what follows. The source code of the program is listed in the appendix for reference.

3.3.1 Surface-lattice representation

In general, a Monte-Carlo simulation for a crystal requires representation of all atoms in a certain volume. In the etching of Si (111) in NH_4F, provided that the etching is of layer-by-layer type, all chemical reaction is limited to sites that are visible from the top. Thus, it suffices to store only the surface atoms in a 2D array of integers, with each integer representing a lattice site. The integer value is the surface level of the topmost silicon atom that occupies the lattice site. This array is sufficient to describe the whole Crystal at any given time. All the necessary related information such as the site species and population statistics can be deduced from this array. However, a rather high computational cost is incurred in extracting this information. To facilitate this aspect of the computation, another integer array—the "position array"—is used to store the positions of surface sites grouped by site species. This array is updated each time an atom is etched.

3.3.2 The etch algorithm

Before the simulation is performed, a set of site-specific etch rates is defined. A single parameter, k_i, characterizes the etch rate for all the sites of species i. If there are a total of N_i sites of species i, then $N_i k_i$ sites are assumed to be etched per unit time. Following Hines' convention, the time unit was chosen so that $k_{kink} = 1$. In this chapter, this unit is simply refereed to as time unit. As long as the same set of site specific etch rates are used, this time unit is consistent.

During the simulation, a random number is first generated to determine which site species will be selected for etching; this is done according to the probabilities calculated from the following equation:

$$P_i = \frac{N_i k_i}{\sum N_j k_j}, \tag{3.1}$$

where k_j is the etch rate of species j, and N_j is the population of species j at the moment when the etch takes place. A second random number is then generated to select a particular etch site within the species. A significant amount of computation is required for updating the "position array," which involves updating not only the site species of the atom being etched but also the site species of its neighboring sites, as those species are likely to be changed as a result of the etching of the atom at the chosen site.

3.3.3 Cyclic boundary conditions

Although the program represents only the surface atoms, it nevertheless requires a huge amount of memory. For example, to simulate a 1 μm $\times$ 1 μm area of the surface, about 70 megabytes of memory is required, while simulation of a 5 μm $\times$ 5 μm area requires 1.75 gigabytes. Without a proper treatment of the boundary, the boundary sites would behave like permanent defect sites, and the effects of those defects would propagate into the center after a certain etching time via the step-flow mechanism. By exploiting the symmetry of the crystal lattice, however, it is possible to employ cyclic boundary conditions [24], a strategy which is equivalent to mapping one edge of the simulated area to the opposite edge. The entire silicon surface is treated as a repeating tiling of the same simulated area (Figures 3.4 and 3.5).

Cyclic boundary conditions were employed in Hines' simulation algorithm, but a rectangular simulation area was used [24]. A rectangular area works well when the simulated steps are always parallel to one of the sides, as in the case of surfaces with a miscut toward $< 11\bar{2} >$ or $< \overline{11}2 >$. For an arbitrary miscut orientation, cyclic boundary conditions are rather cumbersome to deal with, since the intrinsic lattice is lacking in orthogonal symmetry, which often results in a vertical shift, as shown in Figure 3.4. In order to utilize the symmetry of the surface lattice, our program uses a simulation area in the shape of a parallelogram, with its corner angles equal to 60° or 120° (Figure 3.5). This

approach enables us to treat the cyclic boundary conditions in both directions symmetrically, which greatly simplifies the algorithm.

3.3.4 Etching time

In the simulation, the time allocated for the etching of any site in a particular class is assumed to be constant. The actual time it takes to etch away an atom site is calculated as the inverse of its etch rate. Since the total etch rate of species i at any moment is $N_i k_i$,

$$\Delta t = \frac{1}{N_i k_i}. \tag{3.2}$$

After the removal of each atom, Δt is added to the total accumulated etching time. The total etching time obtained in this way is given in units of the simulation time unit. As long as the site-specific etch rates used in the simulation are constants, the cumulative etching time has a constant time unit, which means the etch times for different sessions can be directly compared. The etch time used in the simulation can also be compared to experimental measurements, in which case the actual value of the time unit can be determined empirically—and, consequently, a set of site-specific etch rates in real time units can be derived.

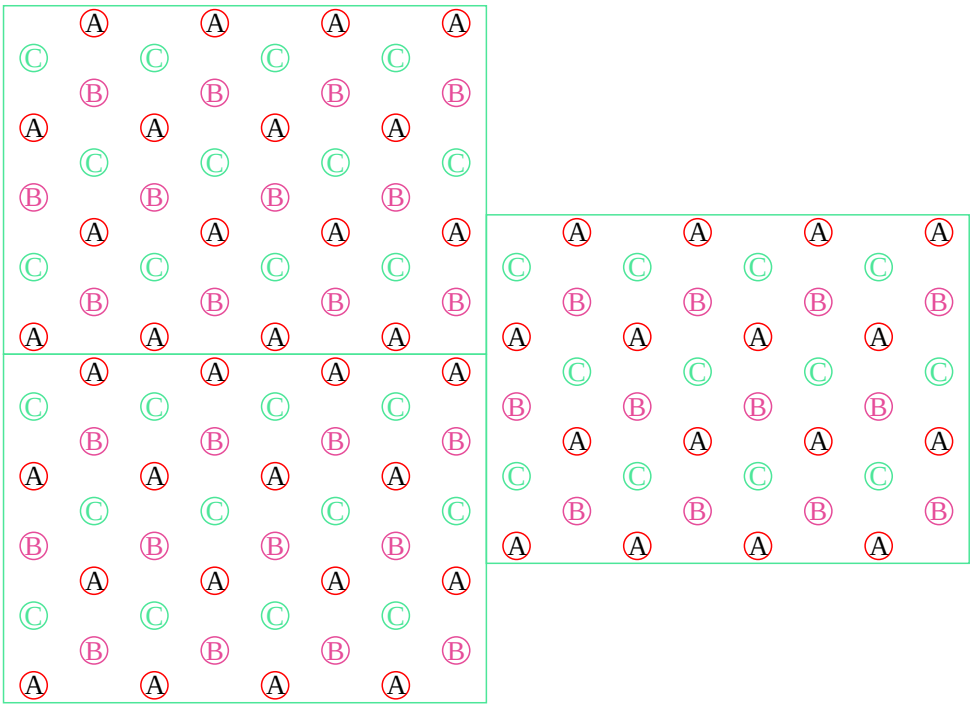


Figure 3.4. Illustration of boundary conditions with a rectangular simulation area. The boundary conditions essentially requires the lattice sites at one edge matches the lattice sites at the opposite edge. Generally, because there are steps across the surface, the surface at two edges are often at different surface layer. As the result, the boundary conditions often require lattice site A match lattice site B or C at the opposite edge. Due to the three-fold symmetry on Si (111), a rectangular simulation area can easily be tiled in one direction. The other direction, however, often requires a slight vertical shift to match the corresponding lattice.

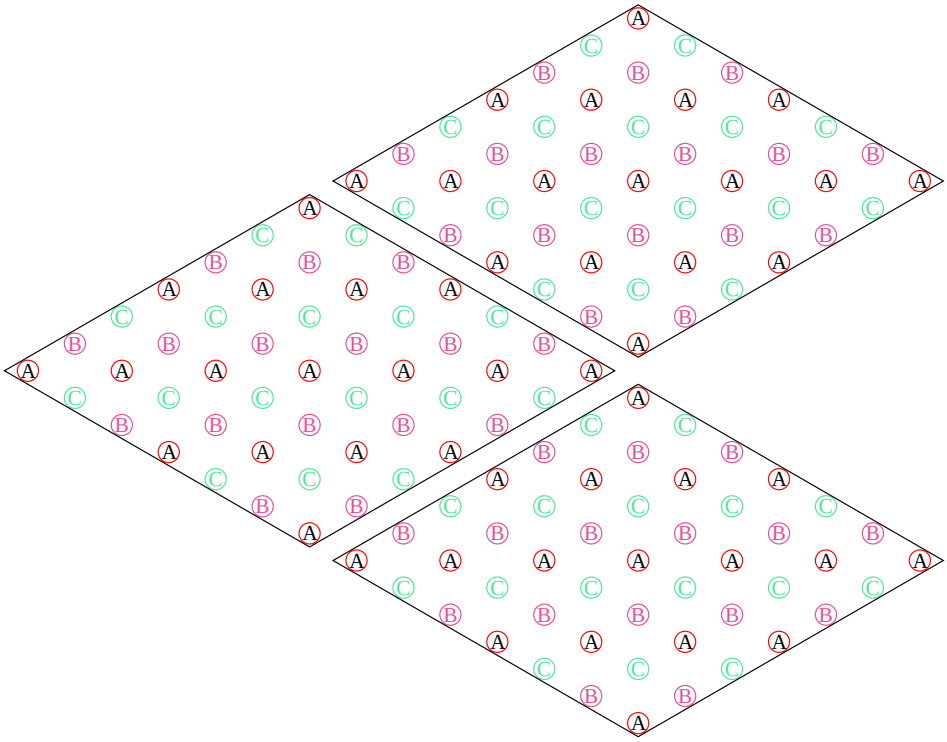


Figure 3.5. Illustration of boundary conditions with a simulation area in the shape of a parallelogram. Refer to Figure 3.4 caption for explanation on the boundary conditions. The chosen parallelogram utilizes the symmetry of the silicon surface lattice, thereby making it much easier to satisfy the boundary conditions.

3.4 The etching process studied by kinetic Monte-Carlo simulations

In this section, kinetic Monte-Carlo simulations will be used to illustrate certain aspects of the development of the surface morphology of Si (111) etched in aqueous NH_4F. The simulation results were obtained using the program listed in the appendix. This program has successfully reproduced the results published by Hines [55, 36]. By comparing the morphology obtained from their simulations and experiments, they suggested that the following etch rates for Si (111) surfaces etched in 40 % NH_4F at room temperature are optimal:

$$k_{kink} = 1 \tag{3.3}$$

$$k_{pt} = 0.1 \tag{3.4}$$

$$k_{di} = 0.01 \tag{3.5}$$

$$k_{mono} = 5 \times 10^{-4} \tag{3.6}$$

$$k_{terr} = 10^{-7} \tag{3.7}$$

Infinity is used for all other species [36]. Here, k_{kink} is the etch rate for kink sites, k_{pt} is for point sites, k_{di} is for vertical di-hydrides, k_{mono} is for step mono-hydrides, and k_{terr} is for terrace mono-hydrides. All rates are given in units of per site per time unit.

The etch rates recommended by Hines et al. are the ones that were used to obtain the simulation results described in the remainder of this chapter.

3.4.1 Step-flow etching

The simulation of the step flow can be directly compared to the results of Ye's experiments [49]. By comparing the step-flow rates obtained in our simulation to Ye's published values, the real value of the time unit can be derived.

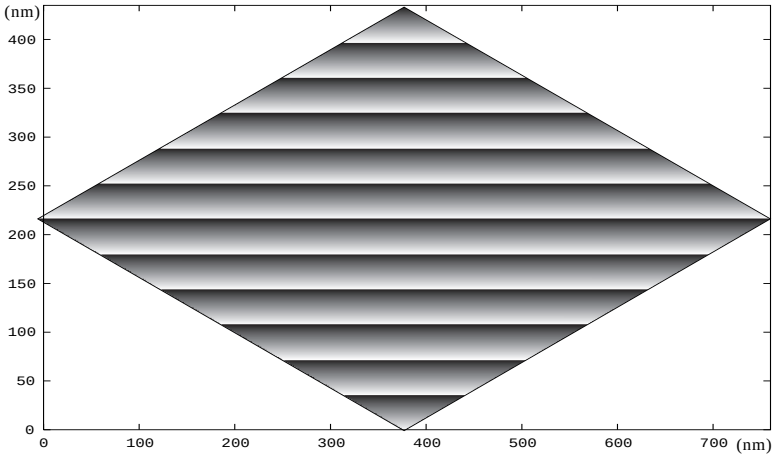
To simulate the step flow, large miscut angles were used, so that there was a limit on the maximum terrace lifetime and the pitting effects were suppressed in both size and number. Figure 3.6 shows the simulation of a surface with a miscut angle of 1.00° toward $< 11\bar{2} >$. The simulation began with a perfectly uniform step-and-terrace morphology, and the etching continued until 6 monolayers (ML) of surface sites were removed. A snapshot of the surface morphology was taken after the etching of every 0.1 ML. For each snapshot, the average position of each step was measured via an edge-detection algorithm. The results are plotted in Figure 3.7. The rate of movement of each step was determined using a linear fit, and the average step-flow rate was computed by averaging the results obtained by the fit for all the steps. This simulation yielded an average step-flow rate of

$$Avg.\,Flowrate = 0.00214 \pm 0.00001 \quad \text{nm/time unit} \tag{3.8}$$

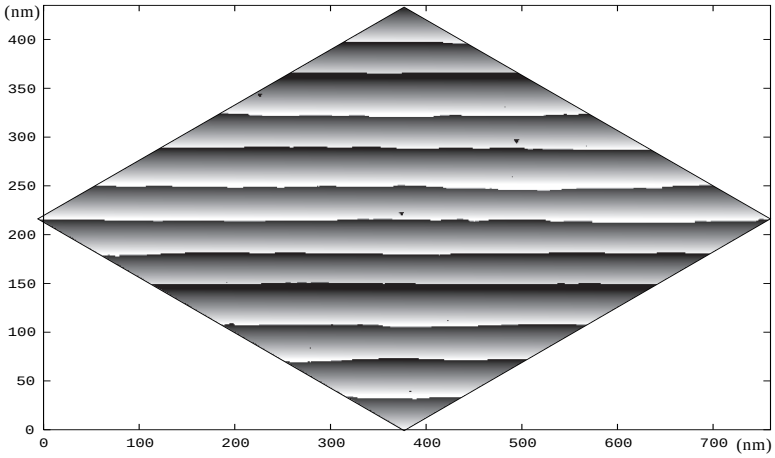
Alternatively, the average step-flow rate can be derived from the time taken to etch away a single monolayer of sites. From equation 2.9, we have

$$L = W/T_w. \tag{3.9}$$

Table 3.1 lists the etching time for each ML along with the step-flow rate calculated from equation 3.9. As shown in the table, the step-flow rates obtained



(a) Start



(b) After 6 ML of simulated etching

Figure 3.6. (a) State of the surface before the simulation was begun. The simulated surface has a 1.00° miscut toward the $< 11\bar{2} >$ direction, which gives an average terrace width of 17.96 nm. (b) State of the surface after 6 ML of simulated etching.

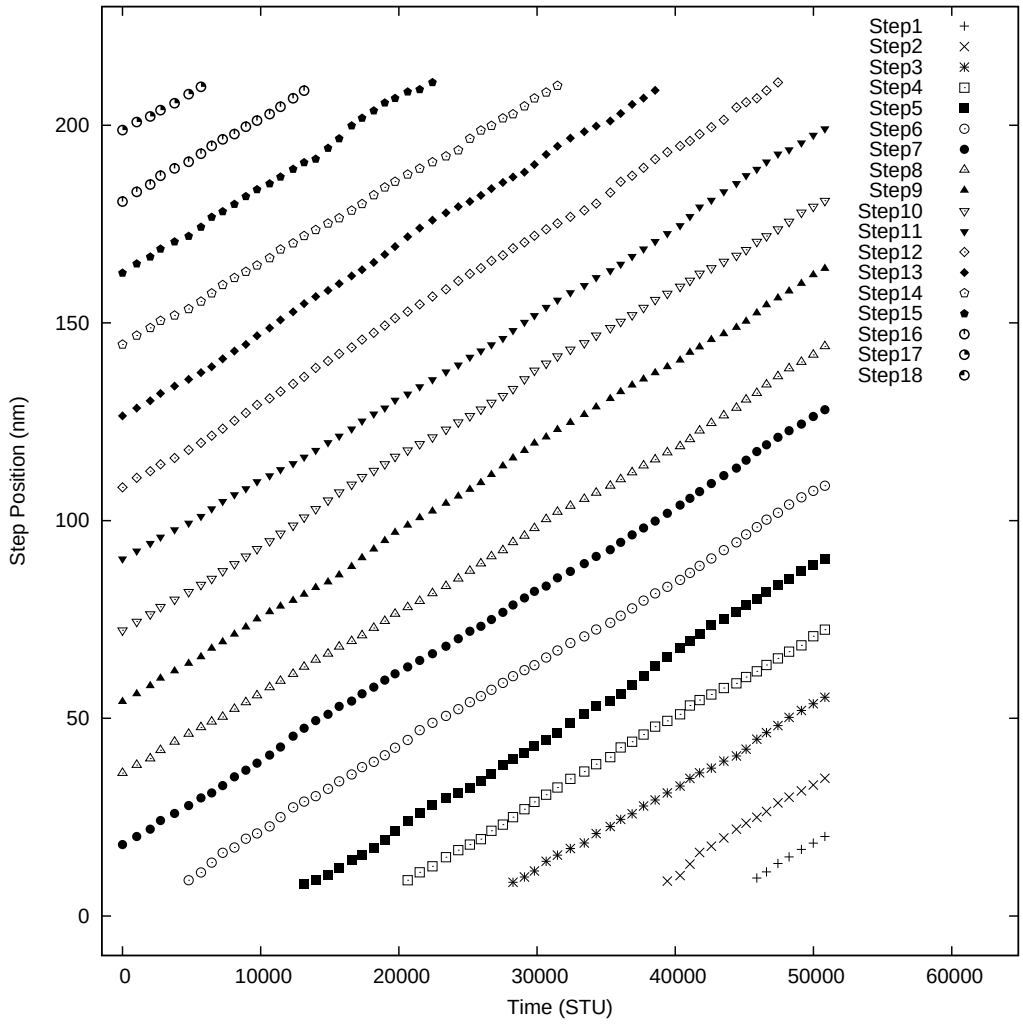


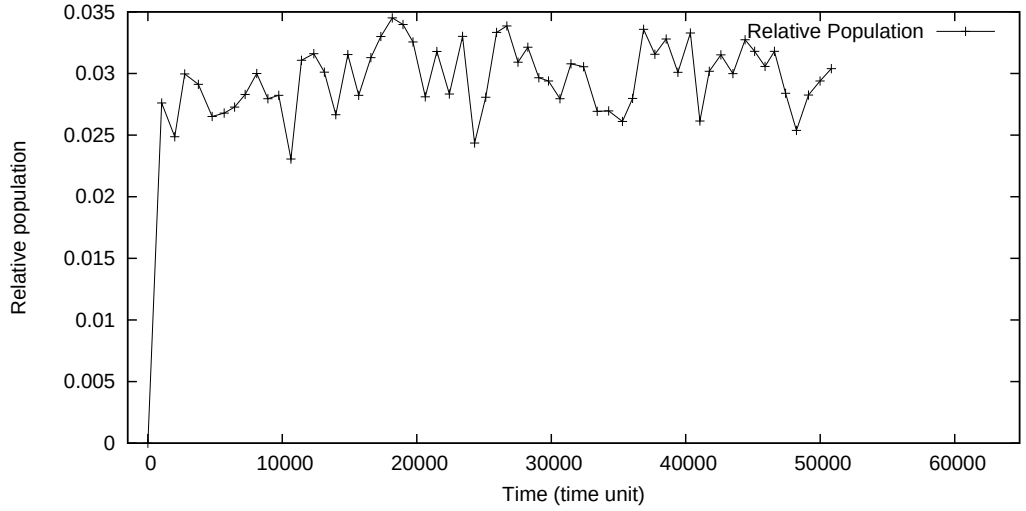
Figure 3.7. Plot of step positions against etching time for the step-flow simulation. By using a linear fit for each step and taking their weighted average, the step-flow rate measured in this plot was found to be 0.00214 ± 0.00001 nm/time unit.

in this way are very close to the values in Eq. 3.8, which was obtained by fitting the positions of each step. Note that the etch time for the first monolayer was longer than that for each of the succeeding monolayers. This can be explained by looking at the plot of populations of kink sites during the simulated etching (Figure 3.8(a)). The simulation starts with an idealized condition, in which the steps are perfectly straight and consist of only step mono-hydrides. As the kink etching is the key mechanism in step etching, it initially takes longer for the steps to be etched, because of the lack of kink sites at that stage of the process. As the simulated etching proceeds, the kink sites start to populate and quickly reach an equilibrium population (3%), and the step-flow rate increases accordingly.

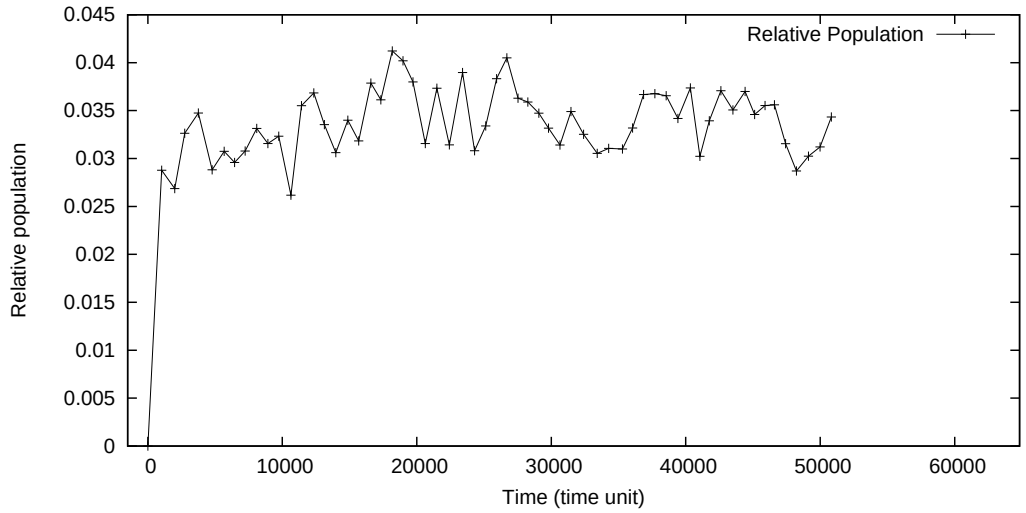
Figure 3.8(b) shows a plot of the relative population of step vertical di-hydrides. Under steady-state conditions, the ratio of the population of vertical di-hydrides to that of kink sites is about 1:1 (with each of these species occupying about 3% of the total number of step sites), which indicates that most of the kinks are confined to a single row.

If we decrease the miscut angle, which in turn will increase the average terrace width, pit etching can no longer be ignored. Collisions between etch pits and steps effectively push the front of the step forward, so for surfaces with smaller miscut angles the apparent average step-flow rate is expected to increase.

Step-flow rates obtained from simulations of surfaces with various miscut angles are plotted in Figure 3.9. The step-flow rates are obtained using equation 3.9. For miscut angles larger than 1°, the apparent step-flow rates are



(a) Kink



(b) Vertical Di-hydride

Figure 3.8. Plots of the population of kink sites and step vertical di-hydride during simulated etching. The population of the specific site class in above figures are shown as the percentage in the total population of the step sites.

ML No.	**1**	**2**	**3**	**4**	**5**	**6**
Etch Time (time unit)	8925.9	8403.9	8603.3	8341.4	8336.6	8213.3
Step-flow Rate (nm/time unit)	0.002012	0.002137	0.002088	0.002153	0.002154	0.002187

Table 3.1. Results of the step-flow simulation

close to a constant value, indicating that the effects of pit etching are minimal. For miscut angles smaller than 1°, the step-flow rates sharply increase as the miscut angle decreases, which indicates an increased contribution from pit etching. Surface morphologies for some of the simulations are shown in Figure 3.10. As the miscut angle decreases, the effects of pitting can be deduced from the morphology, since more pits are formed—and steps deviate further from straightness—as a result of step–pit collisions. The smallest miscut angle simulated in figure 3.9 was 0.02°. For miscut angles smaller than 0.02°, the steps became too fragmented due to sever collision with pits and measurement of average step positions from the morphology became very difficult.

In the simulation results presented above, all the simulated surfaces have a miscut toward $< 11\bar{2} >$. If the miscut is off $< 11\bar{2} >$ orientation, we expect that the kink population will increase, since the steps are no longer facing exactly in

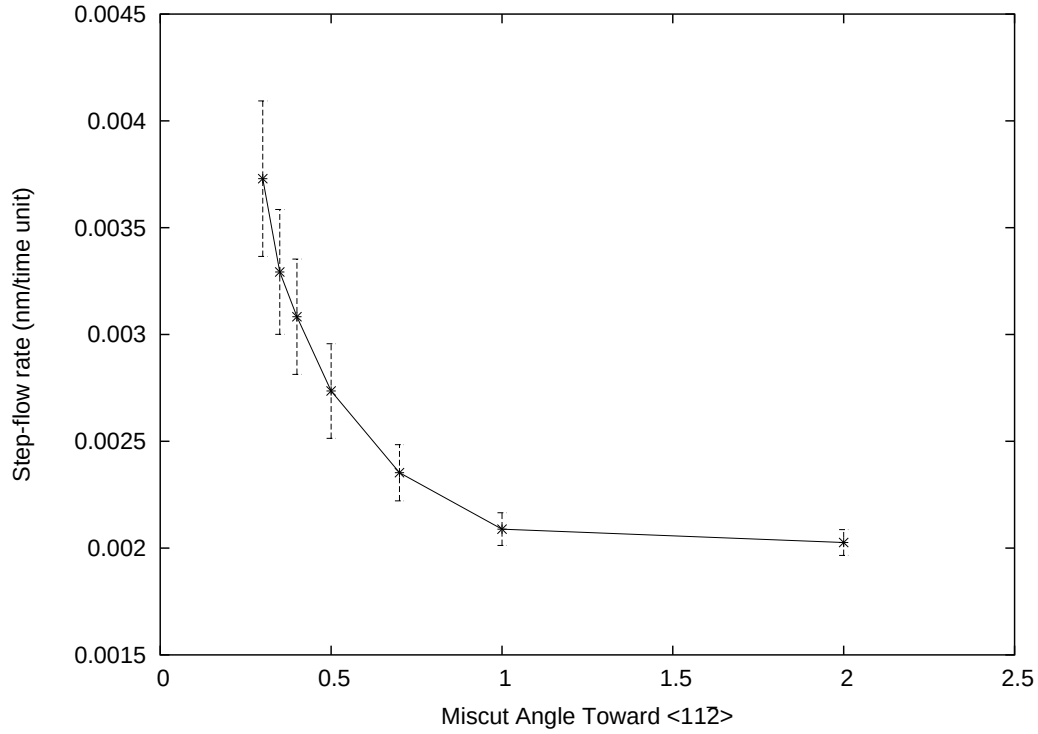
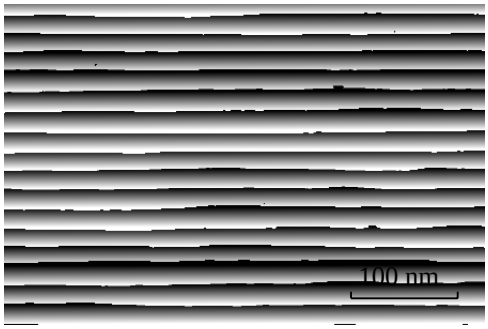
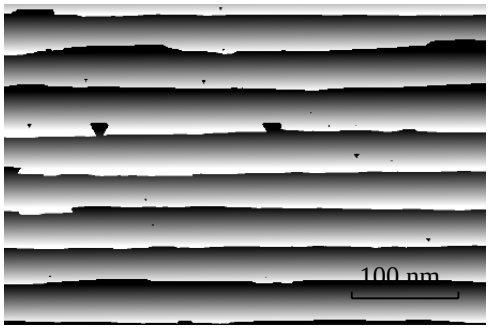


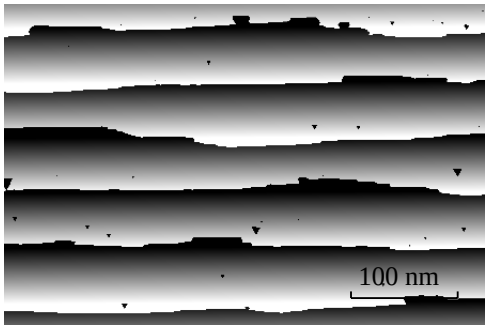
Figure 3.9. Plot of step-flow rates obtained from simulations of surfaces with various miscut angles.



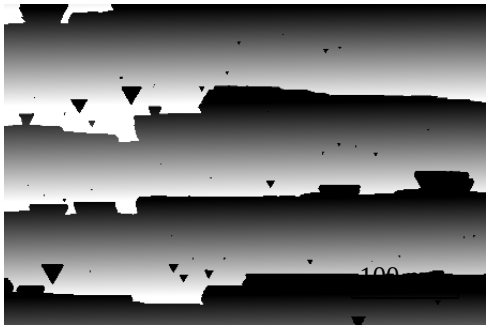
(a) 1.00°



(b) 0.50°



(c) 0.35°



(d) 0.20°

Figure 3.10. Morphology of surfaces with various miscut angles after 10 ML of simulated etching.

the $< 11\bar{2} >$ direction. For purposes of discussion, we will refer to the orientation angle off the $< 11\bar{2} >$ direction as the misorientation angle. Figure 3.11 shows the simulation results for a set of surfaces with the same miscut angle (1.00°) but different misorientation angles. For small misorientation angles (facing the vicinity of $< 11\bar{2} >$), a uniform kink site is distributed along the steps, and as the misorientation angle increases, the kink density also increases. For large misorientation angles (larger than 20°, facing more toward $< \overline{11}2 >$), however, segments that face in the $< \overline{11}2 >$ direction are developed as well as hillocks, which are formed by two straight step segments that face in the $< 11\bar{2} >$ direction and a point site. The average step-flow rates are plotted against the misorientation angles in figure 3.11, the relative site populations are plotted in figure 3.12, and the states of the surfaces after 10 ML etching are shown in figure 3.13. For misorientation angles below 10°, the step-flow rate sharply increases with the misorientation angle. However, for large misorientation angles, the step-flow rate becomes less sensitive to the misorientation angle. For misorientation angles larger than 30°, the step-flow rate actually decreases slightly as the misorientation angle increases. The actual simulation values are shown in figure 3.11. This coincides with the kink population curve, which reinforces the concept of kink-flow etching. On the other hand, the population of vertical di-hydrides consistently increases with the misorientation angle, so the plot displays a monotone curve. For surfaces with small misorientation, the population of vertical di-hydrides is approximately equal to the population of

kink sites, indicating that most of the kinks are confined to a single row. For misorientation angles larger than 10°, the vertical di-hydride population continues to increase but the kink population stabilizes, indicating the formation of di-hydride step segments that are facing in the $< \overline{11}2 >$ direction. For surfaces with a misorientation angle of 60°, the steps are actually facing in the $< \overline{11}2 >$ direction, and the vertical di-hydrides occupy more than half of all the step sites.

In Ye's experiments, the step facing in the $< \overline{11}2 >$ direction was found to have a step-flow rate 3 times larger than that of the step facing in the $< 11\bar{2} >$ direction. The rates were $L_{<11\bar{2}>} = 28$ nm/sec and $L_{<\overline{11}2>} = 86$ nm/sec. From our simulation results shown in figure 3.11, we obtained $L_{<11\bar{2}>} = 0.0021$ nm/time unit and $L_{\overline{11}2>} = 0.0063$ nm/time unit, which agrees with Ye's experimental results. By comparing the step-flow rate for steps facing in the $< 11\bar{2} >$ direction obtained from our simulation (0.0021 nm/time unit) to Ye's result (28 nm/sec), we have

$$1 \text{ second} = 222 \text{ time unit.} \tag{3.10}$$

From equations 3.3–3.7, we derive the following site-specific etch rates in units of per site per second:

$$k_{kink} = 222 \tag{3.11}$$

$$k_{di} = 2.22 \tag{3.12}$$

$$k_{pt} = 22.2 \tag{3.13}$$

$$k_{mono} = 0.111 \tag{3.14}$$

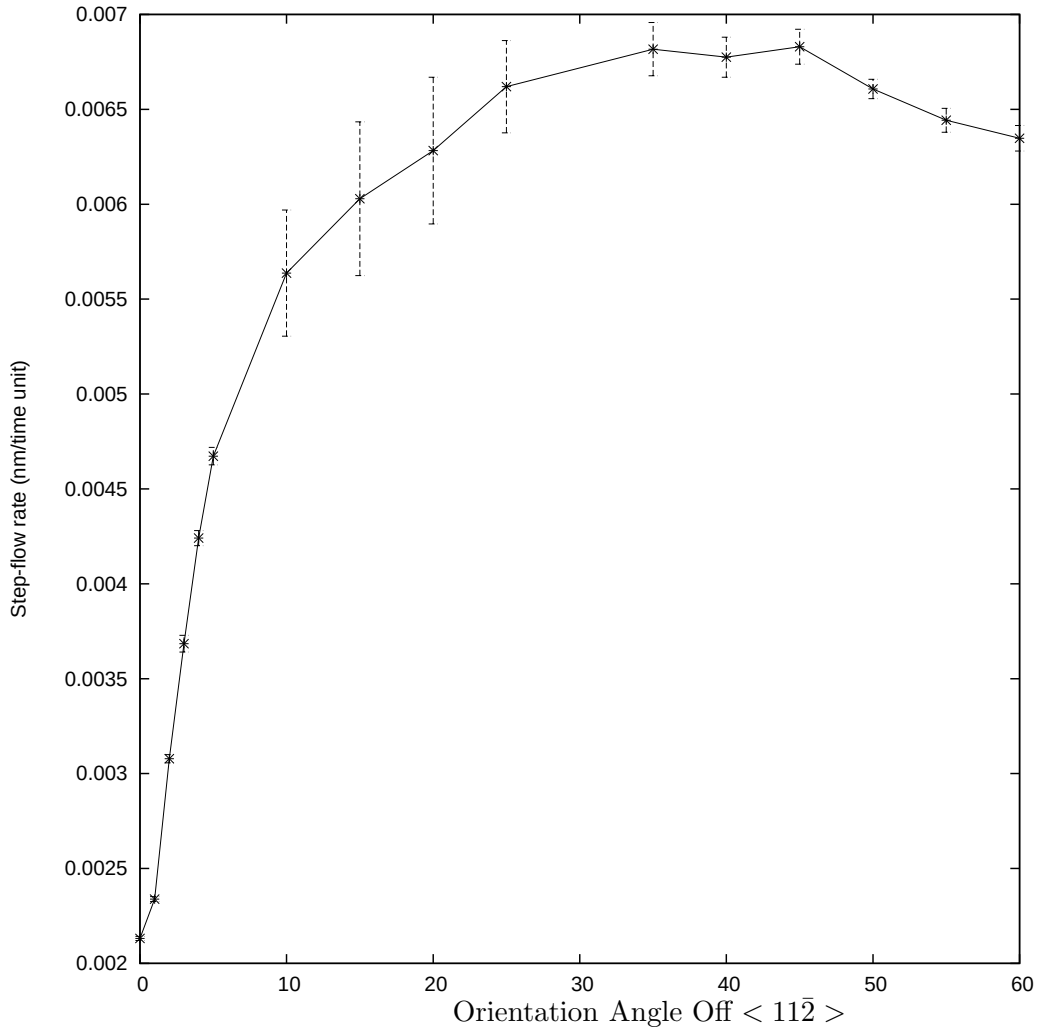
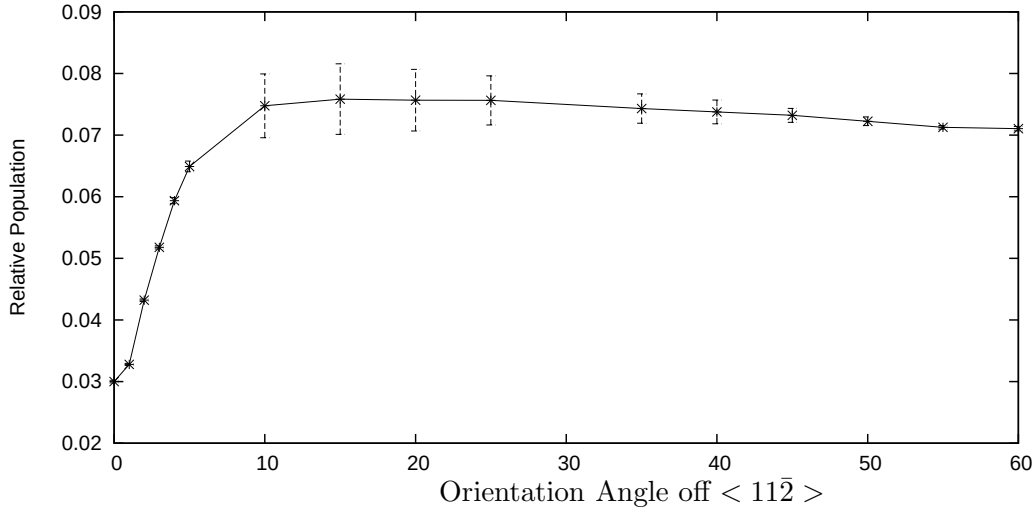
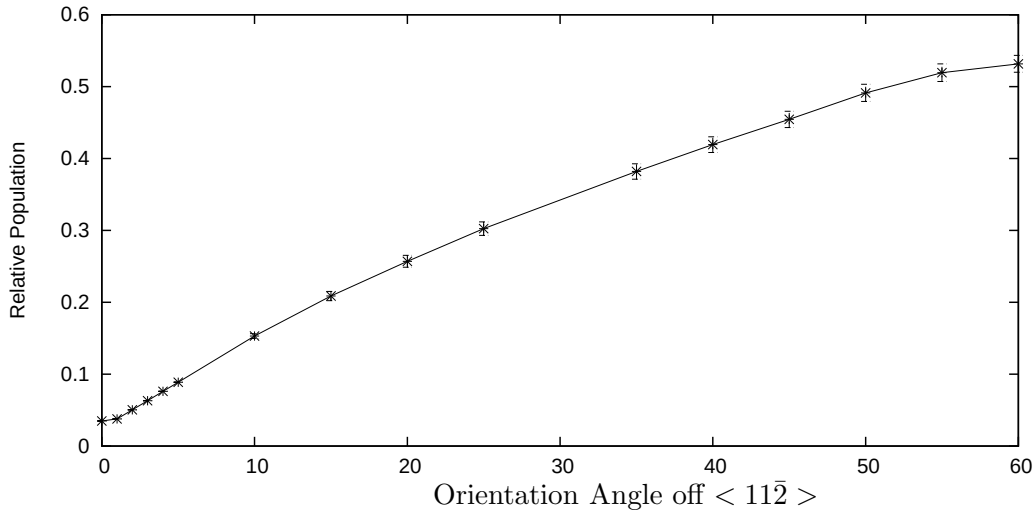


Figure 3.11. Plot of step-flow rates obtained from simulations of surfaces with a 1° miscut angle toward various miscut orientations. The orientation angle in the plot uses the $< 112 >$ direction as a reference.

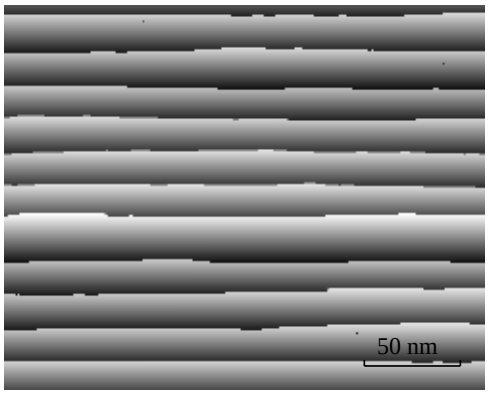


(a) Kink

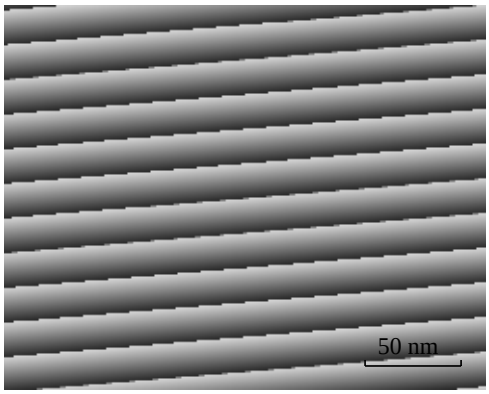


(b) Vertical Di-hydride

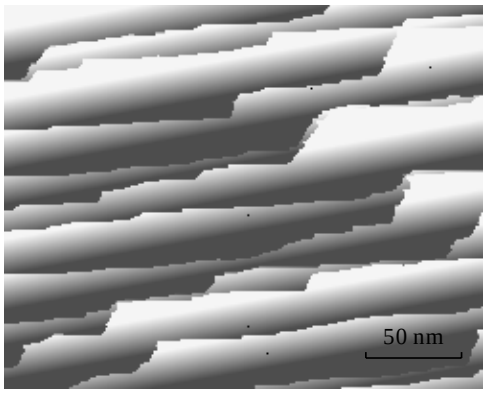
Figure 3.12. Plots of relative populations of kink sites and step vertical di-hydride during simulated etching. The population of the specific site class in the above figures are shown as the percentage in the total population of the step sites.



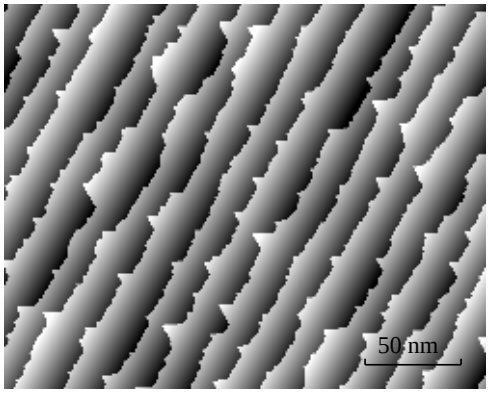
(a) 0°



(b) 5°



(c) 10°



(d) 60°

Figure 3.13. Morphology of surfaces with various miscut orientations after 10 ML of simulated etching. All surfaces have a miscut angle of 1.00°.

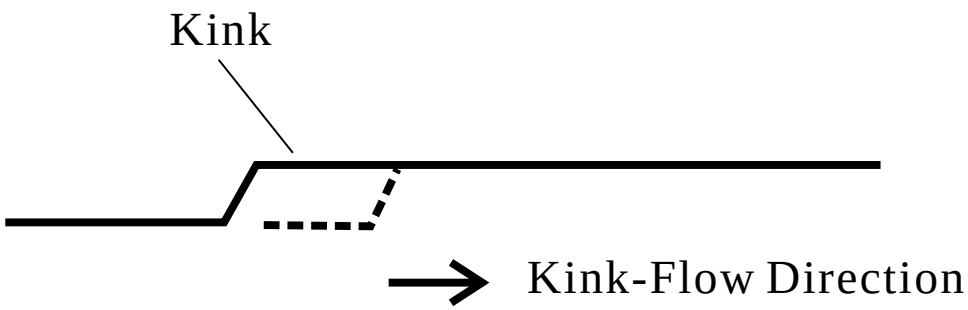
$$k_{terr} = 2.22 \times 10^{-5} \tag{3.15}$$

3.4.2 Pit etching

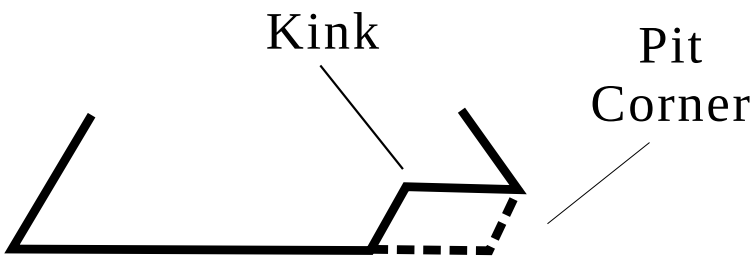
In the previous section, pit etching was mentioned as one of the major components used in determining the surface morphology. In this section, we will use kinetic Monte-Carlo simulation to study the pit growth curve.

A typical etch pit observed in experiments is enclosed by three steps facing in the $< 11\bar{2} >$ directions. For small pits, the steps are contained within a small volume, and the random kink sites generated during etching can more easily be quenched at the corners (Figure 3.14), so we expect that the kink sites will occupy a smaller percentage of the total number of step sites than they would in the case of long-range steps. After the pits get bigger, the longer step segments provide more room for kink propagation, and the kink growth rate is expected to increase.

The simulation starts with the initial placements of a pit of a certain size on a flat terrace with no miscut. In order to obtain the pit growth rate from the simulation, new pits are not prevented from originating. Consequently, after the etching proceeds for about 40 seconds, many new pits are present on the terrace and the center pit start to collide with the new pits, which makes the measurement of the pit growth rate at later stages very difficult. To get around this problem, for growth of pits of a certain size, we used a simulation with a pit



(a)



(b)

Figure 3.14. (a) The kink-flow on a long-range step. There is no natural boundary for the step. The kink site keeps propagating along the step until it collides into another kink. (b) The kink-flow on a pit border step. The step is enclosed by two corners. The kink site disappears once it hits the corner.

that was initiated when it was only a few nanometers smaller than the size of interest, and the etching proceeded until new pits started to grow. For each point, the simulation was repeated 20 times in order to obtain a meaningful average rate and standard deviation. In order to make a direct comparison with the results of the simulations on step-flow, in the following discussion, the pit size was measured from the center to one of its sides, and the pit growth rate was measured as the step-flow rate for each side of the pits.

In Figure 3.15, the pit growth rate is plotted against the pit size, and in Figure 3.16 the relative kink and vertical di-hydride populations are plotted against the pit size. Unlike the step-flow simulation, the pit growth rate obtained from the simulation displays a fairly large deviation, which indicates a larger random factor. Overall, the pit growth rate increases significantly as the pit size increases, until the pit size is larger than 10 nm. The pit growth rate for pits larger than 50 nm is about 0.0016 nm/time unit. This value is lower than the step flow rate of 0.021 nm/time unit obtained in earlier sections. Although we expect the pit growth rate will reach the same rate as step-flow as the pit size grow large enough, due to the limit in our simulation window size, this could not be confirmed. For pits of very small sizes, the average growth rate for them are very small. Thus a pit, after its initiation, may stay below a certain size for a long time, which contributes to the apparent dormant time.

In Figure 3.16, the relative site populations are plotted against the pit size. The kink-site population curve is consistent with the growth-rate curve, which

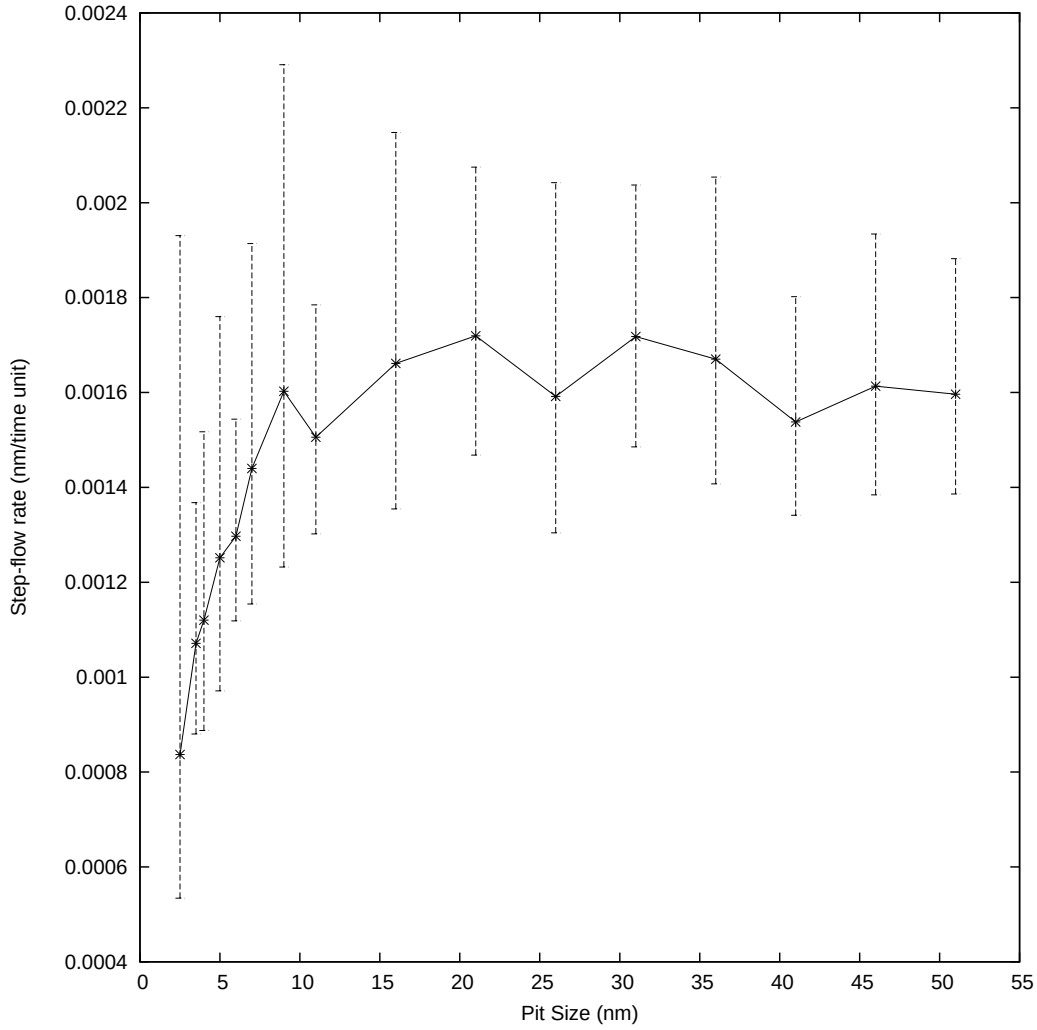
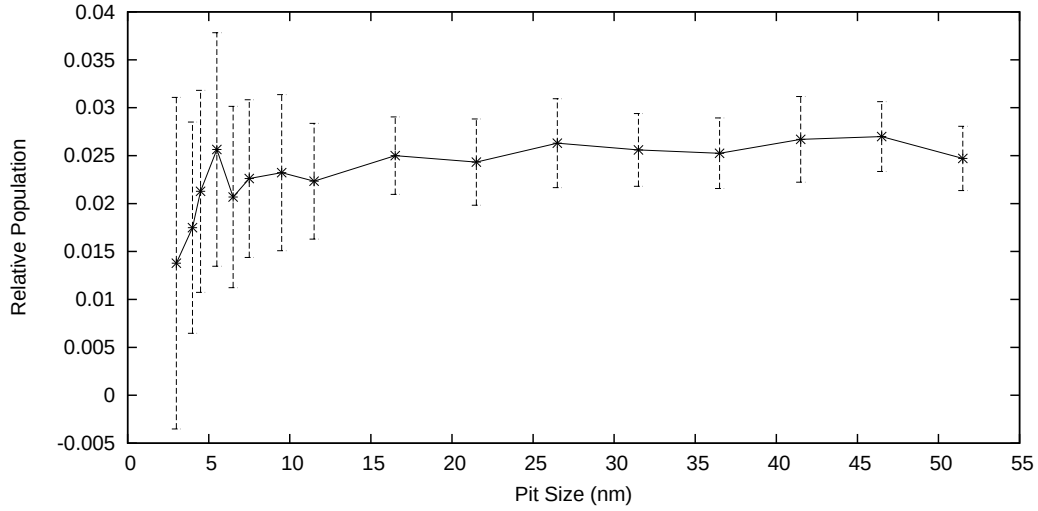
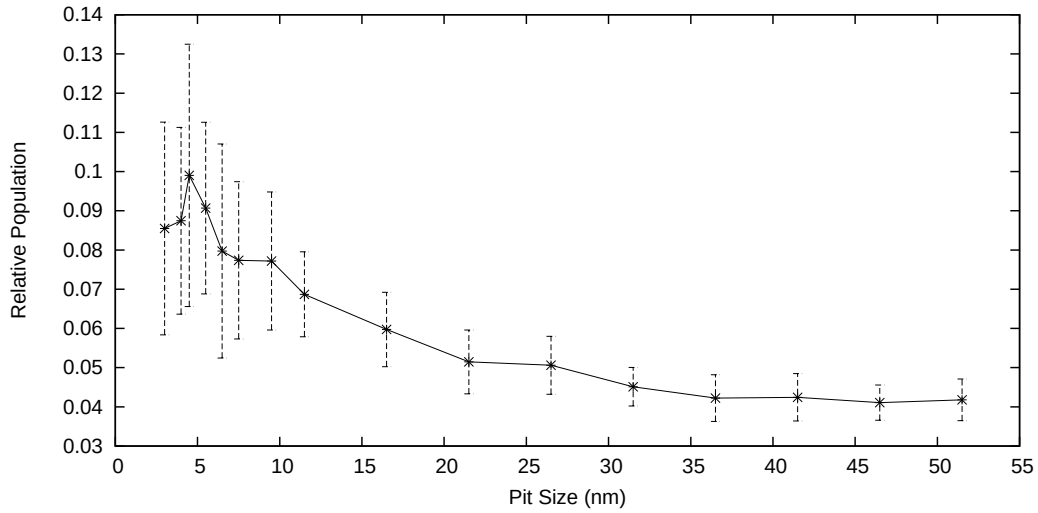


Figure 3.15. Plot of step-flow rates obtained from simulations of pit etching. For pits of very small sizes, the average growth rate for them are very small. Thus a pit, after its initiation, may stay below certain size for a long time, which contributes to the apparent dormant time.



(a) Kink



(b) Vertical Di-hydride

Figure 3.16. Plots of relative populations of kink sites and step vertical di-hydride during simulated etching of pits. The population of the specific site class in the above figures are shown as the percentage in the total population of the step sites.

exhibits a large slope for pits smaller than 10 nm and stabilizes around 2.5% for pit sizes larger than 10 nm. This ratio is less than that for long-range steps, which is around 3% (Figure 3.12(a)). In contrast to the site population we obtained in the simulations on step-flow etching, there is a large population of vertical di-hydrides from the simulations when the interested pits are of small sizes. To understand this result, let us examine the actual atomic site distribution on a small pit. Figure 3.17 is a zoomed-in image of a small pit from our simulation. The kink sites and vertical di-hydride sites are marked in the image. As we notice, instead of single-row kinks, there are quite a few vertical di-hydride sites at the corners forming segments of straight steps that face in the $< \overline{11}2 >$ directions. These corner segments are a result of the accumulated kink annihilation at the corners. During the pit growth, these corner sites require extra time to etch off. Since the corner sites occupy a larger portion of all the step sites when the pit size is smaller, this partially contributes to slow pit growth for small pits.

3.4.3 Collision of pits with steps

The collision of a pit with a straight step creates two corners that connect part of the original pit to the step. At the actual vertex of each of these corners, there is a point site. These corners are etched faster than the rest of the step, and the steps slowly get straighter as the etching proceeds. These effects are commonly

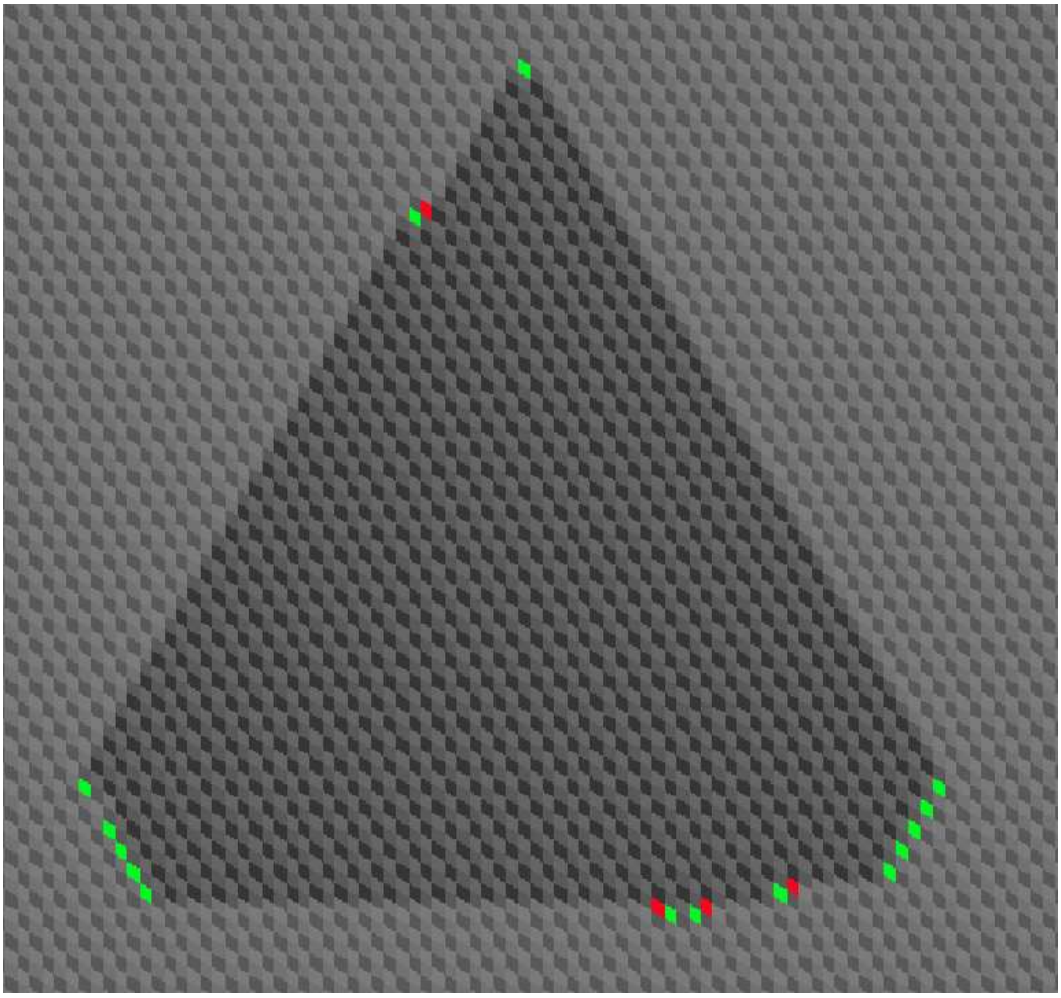
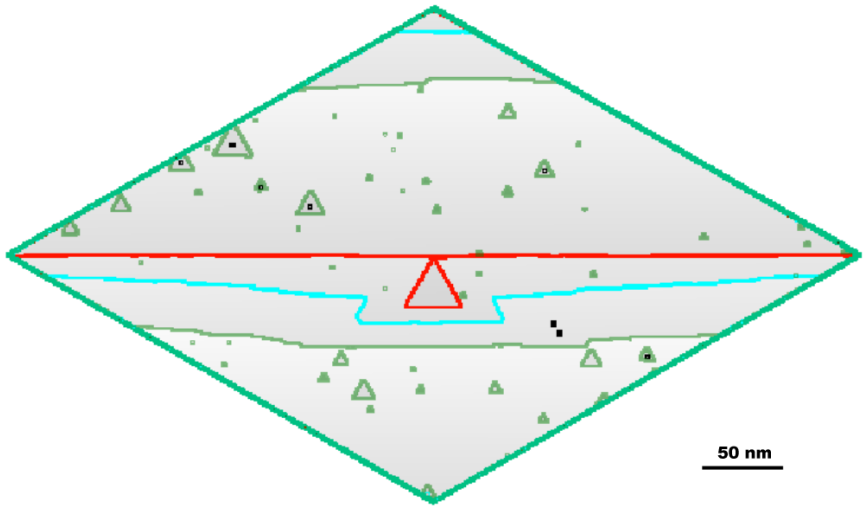


Figure 3.17. Amplified image showing a pit in simulation. The vertical di-hydride sites are marked in green, and the kink sites are marked in red.

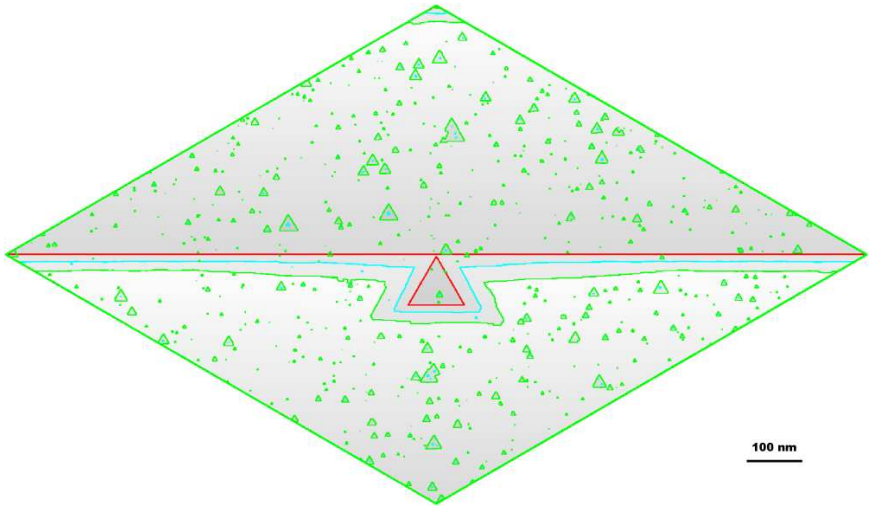
observed in experiments and play an important role in the development of the surface morphology. To simulate these effects, we started with a surface that had a single step in the middle, and we placed pits of various sizes near the front of the step. Figure 3.18 shows a simulation of how the surface morphology evolved after the collisions had occurred.

Figure 3.18(a) shows the collision of a step with a pit that is 10 nm in size. The step returns to a straight shape in 20 seconds, during which time it takes a jump forward. Figure 3.18(b) shows the collision of a step with a pit that is 30 nm in size, in which case the corners persist along the direction of step flow and then gradually become rounded.

The result demonstrates the effect of etch pits on the shape of steps. On surfaces with relative large miscut angles, due to the short terrace life time, the pits are limited to small sizes. As figure 3.18a shows, the steps recover to straight steps after collisions with small pits very quickly. Although these collisions effectively increase the apparent step-flow rate, the steps are maintained straight all the time. For surfaces with smaller miscut angles, the pits can grow into larger sizes, and the steps take longer time to recover from these collisions. Thus the steps on these surfaces tend to become roughened or fragmented.



(a) Pit Size 10 nm



(b) Pit Size 30 nm

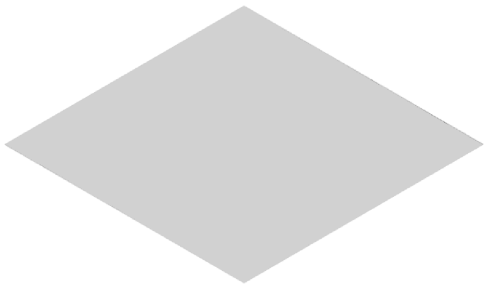
Figure 3.18. Simulation of the etching in the vicinity of a pit–step collision. In above figures, red lines mark the initial step profiles, blue lines mark the step profiles after 10 seconds of etching, and green lines mark the step profiles after 20 seconds of etching. (a) Collision of a pit of 10 nm in size and a straight step. (b) Collision of a pit of 30 nm in size and a straight step.

3.4.4 Etching on perfect planes

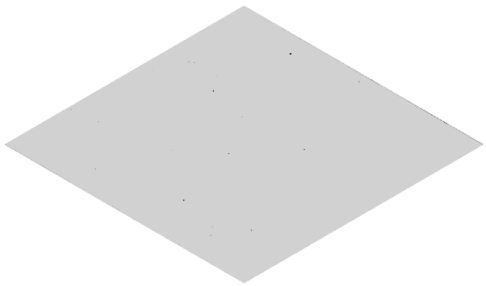
A perfect (111) plane refers to an atomically flat surface that exactly coincides with the (111) crystal plane. For a hydrogen-terminated silicon surface, it is ideally mono-hydride terminated. Such a surface does not exist in reality, but it is the simplest situation to simulate using a computer. Figure 3.19 shows a series of simulation results for a perfect plane, which are rather interesting. Instead of a stable surface morphology, the surface oscillates between being flat and being rough. For these results, we have converted the etching times to real units using equation 3.10.

The surface remained largely flat for as long as 40 seconds, after which pits started to show up and grow. The surface quickly roughened as the pits started to collide with one other, which occurred after about 80 seconds of etching time. Beyond 120 seconds of etching time, the surface returned to being flat, with only a few islands or pits left, and then the cycle was repeated. The pits remained fairly small throughout the entire cycle; once they reached a certain size, they all proceeded to collide with one another, which rapidly gave way to pit annihilation.

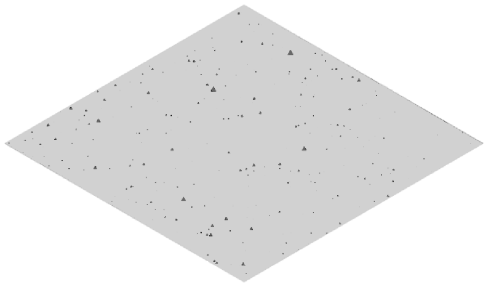
The cyclic morphology observed in these simulations is in direct conflict with our experimental observations on the etching of samples with extremely small miscut angles. The cyclic morphology is due to the initial conditions of our simulation: all the sites within a single terrace have the same lifetime, thus they all have the same probability of being etched—and of initiating pits once the



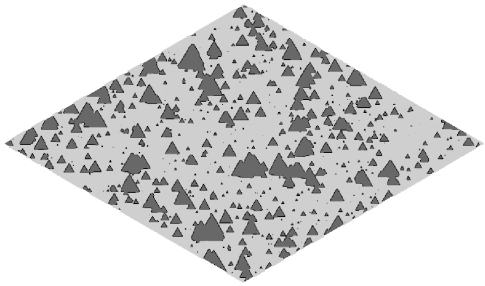
(a) Start



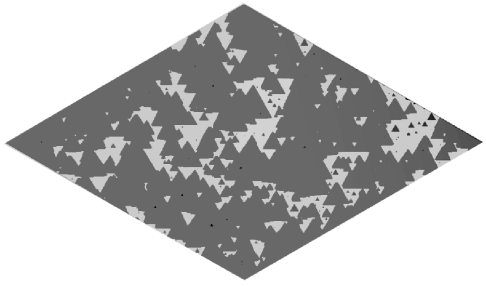
(b) $t = 20$ sec



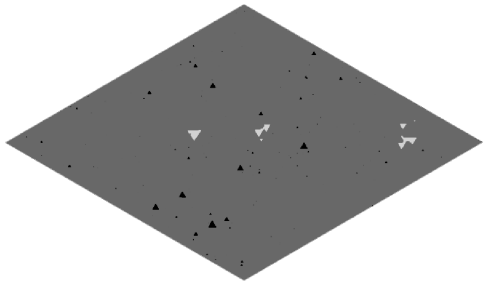
(c) $t = 40$ sec



(d) $t = 80$ sec



(e) $t = 100$ sec



(f) $t = 120$ sec

Figure 3.19. Series of simulations of etching on a perfect plane. The parameters used in this simulation were $k_{kink} = 1$, $k_{di} = 10^{-2}$, $k_{mono} = 5 \times 10^{-4}$, $k_{terr} = 10^{-7}$, and infinity for all the other rates. The time was calculated via equations 3.2 and 3.10. The images are of dimensions 768×444 nm.

etching proceeds. Thus pits almost simultaneously emerge throughtout the terrace. Because of the high density of pits with approximately equal lifetimes, they start to collide with one another once they all reach about the same size. Under real etching conditions, such a near-perfect correlation of pit properties does not exist, since the surface is initially rough. As the surface become smooth, no long range steps exist but rather many short range steps as edges of pits of various sizes randomly distributed across the surface. For a given terrace site, its life time gets reset each time a step sweeps past it. Due to the random distribution of the initial steps, the life time distribution on the overall surface is also quite random. Thus the simultaneous formation of pits will not take place, and is an artifact only shows up in the simulation. As the pits grow, they undergo collisions with pits of different sizes, instead of colliding with pits around the same sizes, or at virtually the same time. Consequently, instead of a sudden annihilation of all the pits, what happens is that large pits form as a result of collisions, or large, zigzagged steps form when large pairs of adjacent pits collide. The formation of these large pits sets the stage for the generation of multiple stacking pits.

3.4.5 Formation of multiple stacking pits

Multiple stacking pits are one of the most interesting morphologies that evolve on surfaces with small miscut angles after long etching times. Figure 3.20 shows an

image from experiments we performed on these features. The sample has a miscut of 0.02°, and it was etched in NH_4F for 3 hours. The stacking pattern consists of about 10 pits of increasing size and approximately a common center. As one proceeds from the inner pits to the outer pits, the inter-pit distance gets smaller. The outer pits are spaced about 60 nm apart, while the inner pits are about 100–300 nm apart.

The formation of stacking pits requires a long etching time and a large surface area, both of which are computationally expensive for the kinetic Monte-Carlo method to simulate. More importantly, only certain sets of initial conditions will lead to the formation of the kinds of stacking pit patterns described herein. Without the appropriate initial conditions, the simulation will develop into a cyclic morphology, as demonstrated in the last section, in which case the etch pits are annihilated before they reach a certain size. In order to partially simulate the development of multiple stacking pits, we started with a series of stacking pits in which the outer pits protected the inner pits from merging with the newly initiated pits, thereby avoiding pit annihilation and allowing the inner pits to get to be fairly large. Figure 3.21 shows the simulation results obtained when the initial pits were of increasing sizes and spaced at intervals of about 30 nm. After 750 seconds of simulated etching, eight new inner pits were formed from the simulation, and they had size intervals that were about the same as those in the original stack pattern. Figure 3.22 shows the results of a different simulation, which started from a stacking pit pattern with size intervals of 66 nm.

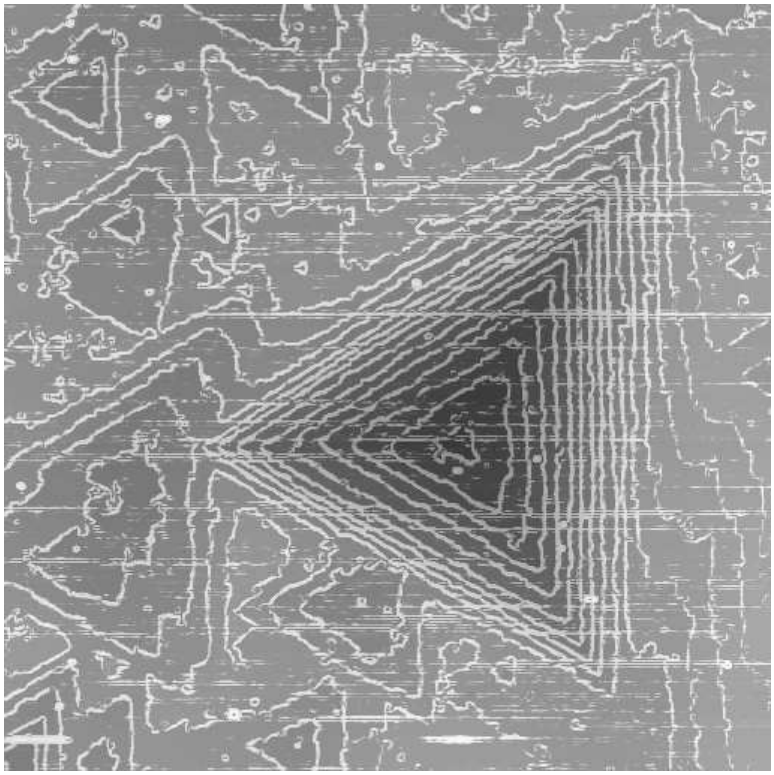
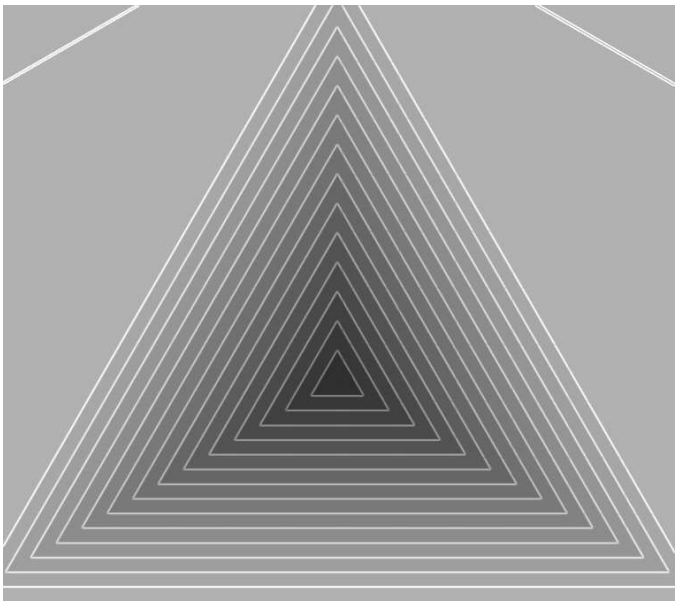


Figure 3.20. A multiple stacking pit pattern developed in experiments

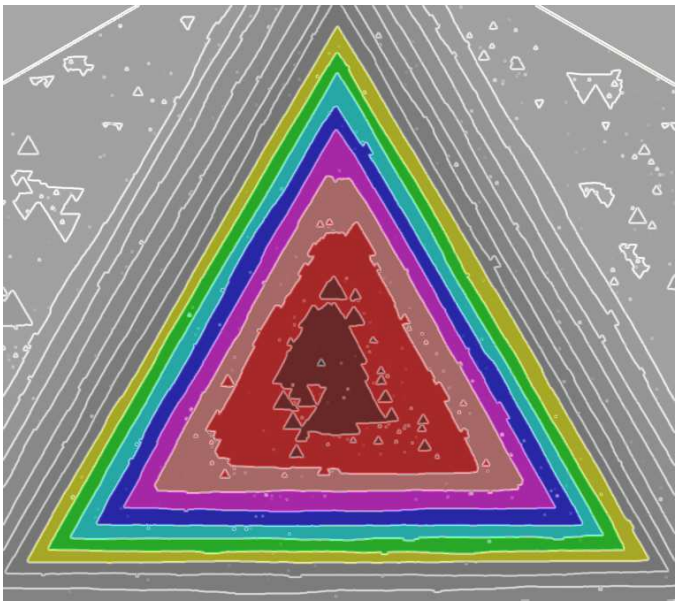
The surface that developed after 85 seconds of simulated etching is shown in Figure 3.22(b), and again the newly generated pits followed the original stacking pattern, this time with an inter-pit interval in the 60 nm range. As these simulations show, the initial pit pattern plays an important role in the newly generated pit stacking pattern.

The results of these simulations can be explained by referring to our earlier simulations on step-flow etching of surfaces with various miscut angles. Because of the effects of pit etching, the step-flow rates for steps with larger average terrace widths were greater than those for steps in which the average dimensions of the terraces were smaller. If we treat the edges of the stacking pits as steps flowing outward, as illustrated in Figure 3.23, then the flow speed of a particular step (A) is positively related to D_A, which is its distance from the next outer step (B). If $D_B < D_A$ (that is, B is closer to *its* next outer step than A is to B), step A will flow faster than step B on average. Thus the result of step flow is a decrease in D_A. And if $D_B > D_A$, there will be an increase in D_A. Thus, steps with unequal spacings will evolve into approximately equal spacings as a result of step flow. This is in agreement with the results of Hines [50]. If this approach is applied to the formation of multiple stacking pits, the inner pits will always grow into a pattern with size intervals approximately equal to that of the outer pits, as shown in both simulations and experiments (figures 3.21, 3.22, and 3.20 respectively).

Although this simulation did not illuminate how the stacking pits get

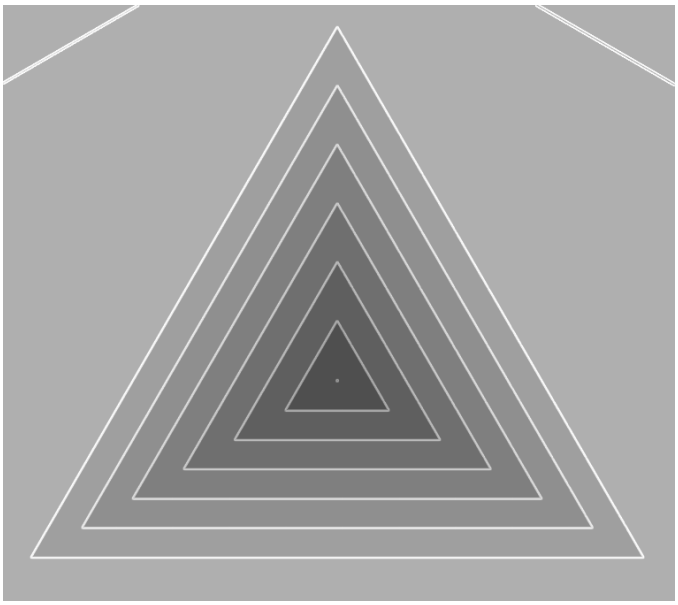


(a) Start

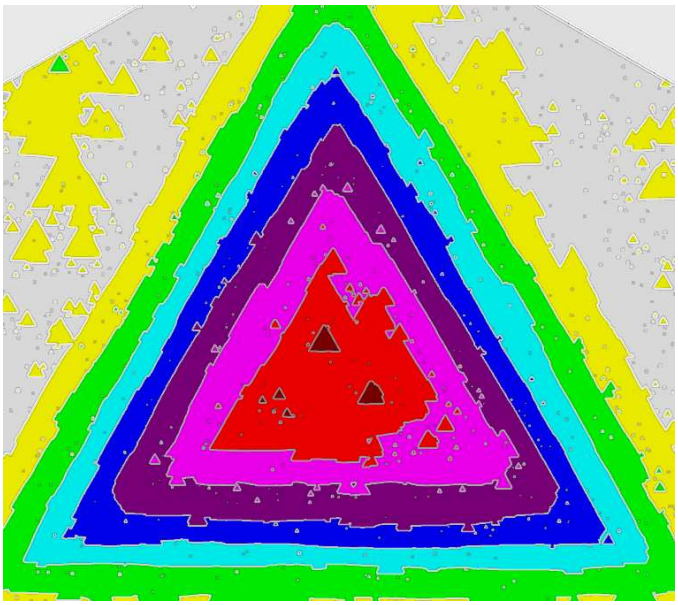


(b) 750 sec.

Figure 3.21. Simulation of etching on multiple stacking pits with a stack interval of 30 nm. The pits shown in clolor are newly initiated by etching.



(a) Start



(b) 870 sec.

Figure 3.22. Simulation of etching on multiple stacking pits with a stack interval of 66 nm. The pits shown in clolor are newly initiated by etching.

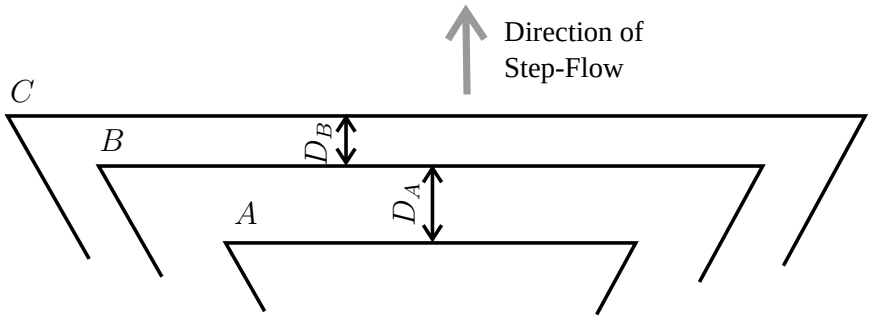


Figure 3.23. Diagram of step flow on steps with unequal step spacings

initiated, it does show how they grow after they are initialized. The direct simulation of these initial pit formations using the current model and algorithm proves to be very difficult due to the limitation of the simulation window size and computation time. In experiments, however, the stacking pits are generated easier than in the simulations, which is an indication that the current model is over simplified. This will be re-examined in chapter 4.

3.5 The limits of kinetic Monte-Carlo simulations

Kinetic Monte-Carlo simulations are an excellent tool for linking the atomic-level reactivity to surface morphology. By comparing the simulated results with experimental measurements, a set of site-specific etch rates can be extracted, as demonstrated by the work of Hines [55, 36]. The extraction of site-specific

reaction rates is valuable for the study of etching mechanisms on the atomic level. With an appropriate set of parameters, kinetic Monte-Carlo simulations are also an excellent tool with which to gain insight into the evolution of the surface morphology. As demonstrated in this chapter, such simulations allow us to perform artificial experiments under conditions that cannot be realized in an experimental environment.

The technique does have its limits, however, the major one being a limitation on the simulation window size. The program typically is limited to simulating a surface area that is no larger than a few micrometers in any direction. To simulate larger surface areas, the number of sites involved in the calculations increases quadratically, as will both the memory and the computing time. In order to avoid introducing sites with artificial boundary defects, the simulation imposes cyclic boundary conditions that do not accurately reflect the real etching conditions. Use of cyclic boundary conditions precludes the free evolution of the surface morphology near the boundary, thereby deviating from its physical counterpart. The simulation results can be confidently or accurately relied upon only if the sizes of the features of interest are much less than those of the simulation window. For surfaces with a large miscut angle, the features of interest are often smaller than the width of a narrow terrace, on account of the strong step–step interaction. For such surfaces, site-specific simulation works well. For samples with small miscut angles, however, evolution of the surface morphology often involves observing several micrometers of surface area, as

indicated in our experiments, and the direct study of the surfaces is often beyond the capability of kinetic Monte-Carlo simulations. By clever tuning of the parameters, it's possible to scale certain morphology features and dynamics to be contained within the finite simulation window. However, the underlying SOS model intrinsically is non-scalable, and not physically based tuning of the etch parameters inevitably causes the simulation to deviate from the real physics.

The second limit lies in the simplification of the etching algorithm. To realize savings in computational time and cost, use of these simulations requires quite a few simplifications. For example, the edge terrace mono-hydrides are expected to have greater reactivity than the general terrace mono-hydrides. Because of the "pairing effect" in the step mono-hydrides, the morphological effects of the fast etching of edge terrace mono-hydrides coincides with the fast etching of step mono-hydrides. Thus, ignoring the distinction between the edge terrace mono-hydrides and the step mono-hydrides can be compensated for by using an "apparent" etch rate for the step mono-hydrides, but then the etch rate for step mono-hydrides extracted from the simulation would differ from the "real" etch rate. As another example, the tri-hydride species has been observed in infrared experiments [56], which indicates that its etch rate should be finite, possibly even comparable with that of a "point" site. In the simulation, however, it is assumed to have an infinite etch rate. Fortunately, because it is not directly linked to observable morphology features, this assumption introduces no apparent conflict with experiment. Similar simplifications can be applied to other classes of sites.

Thus, the site-specific etch rates extracted from the simulation should be treated as apparent etch rates. This must be considered when the apparent etch rates are compared with results obtained by other methods.

A third limit comes from the SOS model and the assumption of a perfect crystal lattice. As is well known, defects in crystal lattices are quite common. The silicon samples used in STM experiments are often also heavily doped (the samples used in our experiments were doped at 10^{18} cm^{-3} level). It is commonly believed that the defect sites are a source of pit initiation and that the dopant sites are likely to yield different chemical reactivities. As simplifications of this type are incorporated into the simulation, these effects have to be added to into the apparent etch rates as well.

In the next chapter, we will try to address some of the limitations and directly study the results observed in our experiments using Monte-Carlo simulation.

Chapter 4

Si(111) Morphology Evolution Studied by Kinetic Monte-Carlo Simulation

4.1 Crystal defects and etch pit initiation

In the previous chapter, we discussed the principles of kinetic Monte-Carlo simulation and they were used to study the etching dynamics discussed in chapter 2. The site specific etch rates used by Hines [24, 55, 36] have given good agreement between their simulations and experiments, and were also used throughout chapter 3 so that a reasonable comparison of our results with the references could be made. However, this set of parameters doesn't yield good agreement with our experiments, especially with samples having with smaller miscut angles. A further modification to the original model is required to get better agreement between our simulations and experiments.

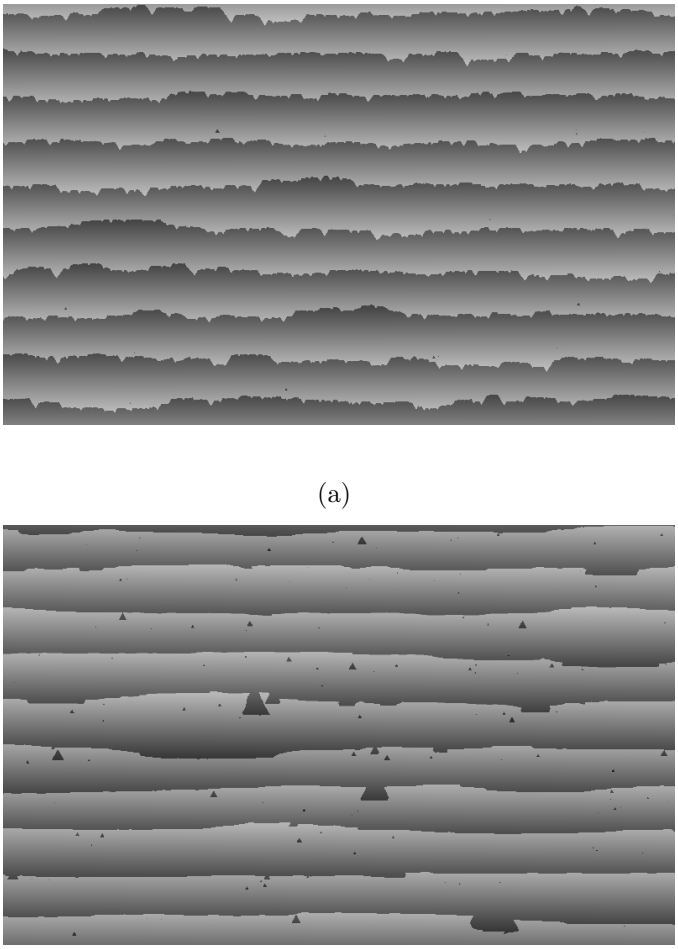
In Hines' experiments, samples with a 0.35° miscut toward either the $< 11\bar{2} >$ or $< \overline{11}2 >$ directions were used. Figure 4.1 shows the simulation results obtained

using our simulation program. The results are consistent with those produced by Hines [55, 36]. The morphology with miscut of 0.35° toward $< 11\bar{2} >$ has straight steps and a few etch pits a few nanometers in size (figure 4.1b). The morphology with the same miscut but toward $< \overline{11}2 >$ has steps distributed with hillocks, and almost no pits are visible (figure 4.1b). This simulated morphology compares very well visually with the surface produced by Hines' experiments.

Our experiments however, use samples with smaller miscut angles. Figure 4.2 shows the observed and simulated morphologies of Si(111) with 0.09° miscut toward the $< 11\bar{2} >$ direction. Figure 4.2b was obtained with the same parameters used by Hines'. The etch pits and step edge roughness are much more pronounced in the simulations than those observed in the experiments. The discrepancy suggests that the etch rate for terrace sites should be smaller than that used in the simulations — 10^{-7}. Figure 4.2c and 4.2d were obtained with $k_{terr} = 10^{-9}$ and $k_{terr} = 10^{-10}$ respectively, which gives much closer agreement with the experiment.

Our experimental environment is quite similar to that of Hines. Although both used 40% NH_4F at room temperature, the simulations indicated a two orders of magnitude difference in the terrace site etch rate, which is very difficult to explain. The conflicting results suggest the original theoretical model requires revisiting and perhaps refinement.

In the original model, etching of a terrace site is the only cause of pit initiation. Although due to the initial nonlinear pit-growth curve, it takes some



(a)

(b)

Figure 4.1. Simulated steady-state etch morphologies of Si(111) surfaces with miscut 0.35° etched in NH_4F. (a) The surface is miscut toward the $< \overline{11}2 >$ direction. (b) The surface is miscut toward the $< 11\bar{2} >$ direction. In both images, the surface has been etched for 10 monolayers.

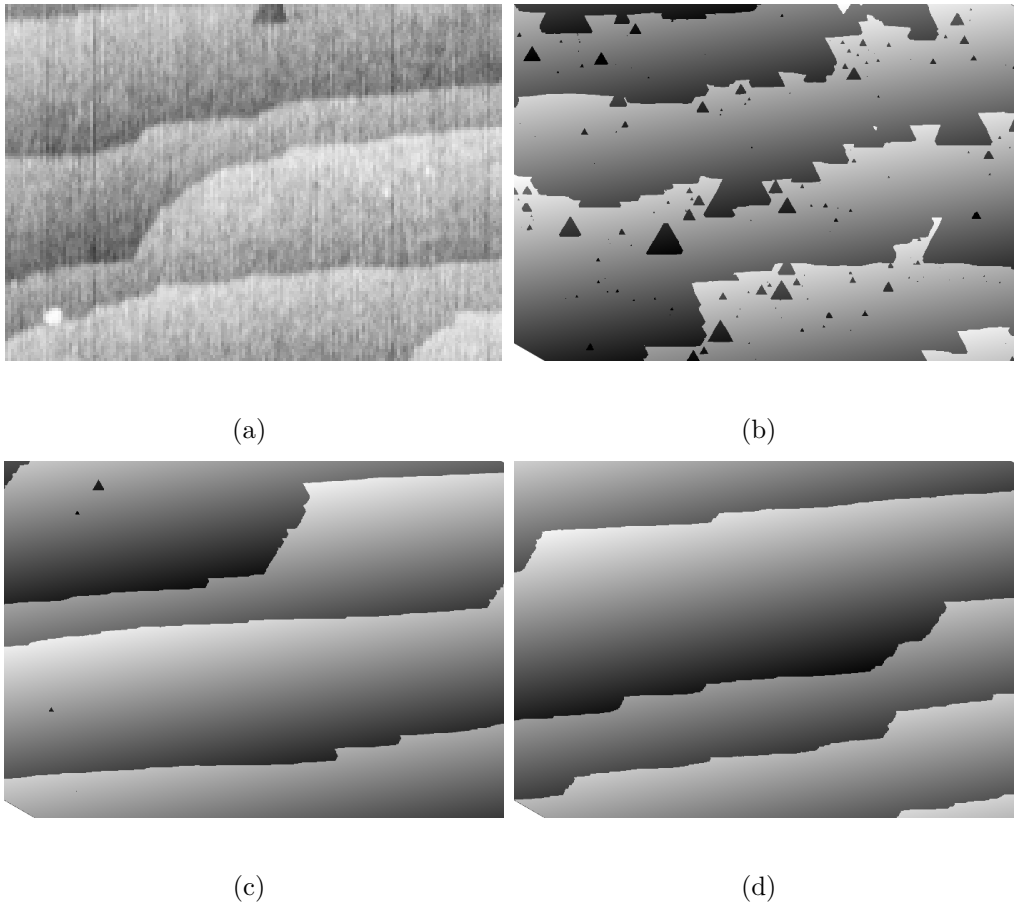


Figure 4.2. (a)Experimental results of steady state etched morphology on a surface with 0.12° miscut. (b) Simulated results obtained with $k_{terr} = 10^{-7}$. (b) Simulated results obtained with $k_{terr} = 10^{-9}$.(b) Simulated results obtained with $k_{terr} = 10^{-10}$. Image size 700 × 500 nm.

time before the pits have grown large enough to observe, the terrace site etch rate still is directly linked to the pit density. The discrepancy between figure 4.2a and 4.2b inevitably leads to a drastically different chemical reactivity for the terrace sites, which is quite unlikely given the similar experimental conditions. We propose that the pit initiation is largely attributed to the surface defect sites, including both the dopant sites and structure defect sites such as dislocation defects. The defect sites are likely to have a much higher reactivity than non-defect terrace sites, and the etching of those defect sites effectively initiates etch pits. Since different researcher efforts use silicon wafers from a variety of sources, they are likely to have different defect distributions, thus they may observe very different pit initiation rates and activities, even though the chemical reactivity for each site class remain the same.

This concept is not surprising as etching has been one of the standard techniques to detect bulk defects in crystals. Although the pits which develop during conventional etching are more akin to macro objects, and the pits discussed in this thesis are more at the atomic scale, and both have different mechanisms and properties, there are in fact quite a few similarities and they may share a similar chemical origin.

4.2 Modified SOS Model

The original SOS model was modified to simulate the above described defect effects.

Two different types of defect sites are distinguished: chemical defects and lattice defects. The former are mostly dopant sites. The dopant atoms share the same chemical bonding structure as the native atoms, and they don't break the lattice structure from an algorithm point of view. Due to the apparent different atomic sizes and bonding strengths, the chemical structure near dopant sites are more stressed than in the pure silicon structure. This structural stress increases the reactivity around the doping sites. This dopant effect in etch rates have been observed by Maher [57] and Ukraintsev [58] The dopant sites are represented by the original model, but now with randomly selected sites and assigned a higher etch rate. Although strictly speaking, the dopant effect is not limited to single dopant atoms, for simplification, the neighboring effect is limited. As a consequence, the dopant etch rate obtained in the simulation will be higher than the actual etch rate, as it includes all the neighboring effects.

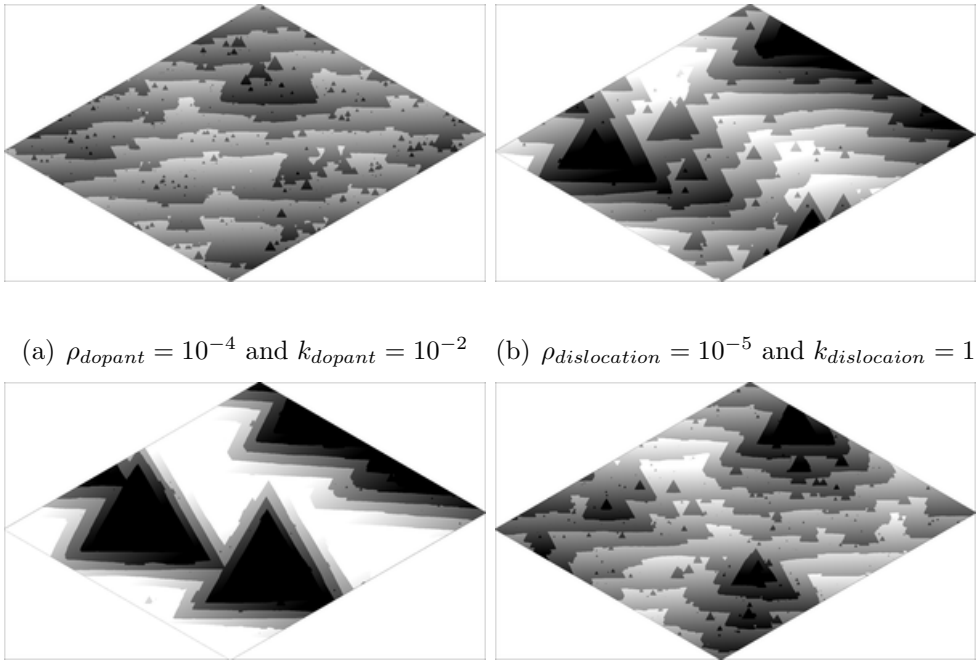
The lattice defect sites are more difficult to treat, since they actually change the lattice structure. Although there is an available model to treat various lattice type defects [59], it is computationally impractical to represent the defects with atomic accuracy. As a simplification, these defect sites are also modeled by randomly selecting sites and assigning them higher etch rates. During etching,

while the distribution of dopant sites should vary for each lattice plane, and the dopant sites should redistribute for each layer, the lattice defect sites are expected to persist for a few layers.

In the modified algorithm, the simulation accepts site concentrations for both dopant sites and lattice defect sites. Both site classes have higher etch rates while the etch rate for terrace sites are reduced to $k_{terr} = 10^{-12}$. Before the etching starts, both dopant and defect sites are randomly selected. During etching, with each removal of a dopant site, a new dopant site is randomly selected. With the removal of a defect site, the defect is propagated to the next lattice layer.

To illustrate the effectiveness of the model, figure 4.3 shows the simulation results obtained for various defect distributions.

Since the dopant sites are randomly distributed for each atomic layer, the surface morphologies resulting from the dopant etching are very similar to the morphology resulting from the simulation of a no-dopant model with a high terrace site etch rate (figure 4.3a). In the etching algorithm, the program selects a site to etch according to the relative etch rate $R_j = k_j \times N_j$. Thus, as long as $R_{terr} = k_{terr} * N_{terr}$ from a no-dopant model is equal to $R_{dopant} = k_{dopant} * N_{dopant}$ from the dopant model and with negligible terrace etch rate, the two models are mathematically equivalent. However, physically, both k_{terr} and N_{terr} should be constants for certain chemical conditions, while both k_{dopant} and N_{dopant} can be variables for different surfaces according to the dopant density and dopant element.



(a) $\rho_{dopant} = 10^{-4}$ and $k_{dopant} = 10^{-2}$ (b) $\rho_{dislocation} = 10^{-5}$ and $k_{dislocaion} = 1$

(c) $\rho_{dopant} = 10^{-5}$, $k_{dopant} = 10^{-3}$, $\rho_{dislocation} = 10^{-5}$ and $k_{dislocaion} = 1$

(d) $\rho_{dopant} = 10^{-4}$, $k_{dopant} = 10^{-3}$, $\rho_{dislocation} = 10^{-5}$ and $k_{dislocaion} = 1$

Figure 4.3. Simulated steady state morphology for surfaces with 0.09° miscut and misorientation of 2°. (a) etched with only the dopant effect, (b) etched with only the dislocation defect effect, (c) the amount of dopant random etching renders the formation of stacking etch pits more pronounced, (d) increased dopant random etching counters the formation of stacking etch pits.

N_{dopant} can be determined from the doping density of the silicon wafer, which can be roughly determined from the resistivity. For a doping density of 10^{18} cm^{-3}, nominally 10^{-5} of the total sites are dopant sites. Since about half of total sites are terrace sites, the etch rate for the dopant, k_{dopant} has to be $5 \times 10^4 k_{terr}$ to produce the same morphology as before. Since $k_{terr} = 10^{-9}$ matches our experiments more closely, a reasonable value for k_{dopant} is 5×10^{-5}.

For dislocation defects, their concentration is expected to be much lower than that of dopant, and also sample dependent. Common types of structure defects include vacancies, interstitials, and dislocation defects, which include edge dislocations and screw dislocations. Vacancies and interstitials also only occupy single lattice planes and their behavior may be included using the dopant model. Dislocation defects, however, propagate during etching. The edge dislocation propagates laterally, and the screw dislocation propagates vertically into the bulk. From a macroscopic point of view, these types of dislocations persist during etching. The existence of the dislocation defect sites, causes certain sites to have a higher probability to form new pits, which may lead to the pit stacking formation that we observed in the experiments (figure 4.3b).

As was discussed in section 3.4.5, the formation of multiple stacking pits is attributed to both repetitive pit initiation and step flow, which in turn is related to random etching. Figures 4.3c and 4.3d show this effect. With a certain amount of random etching attributed to the dopant etching, the effects of multiple stacking pits become more pronounced than without random dopant

etching (figure 4.3c). By further increasing the dopant etching effects, the formation of stacking pits is reduced.

4.3 steady state morphology dependence on wafer miscut

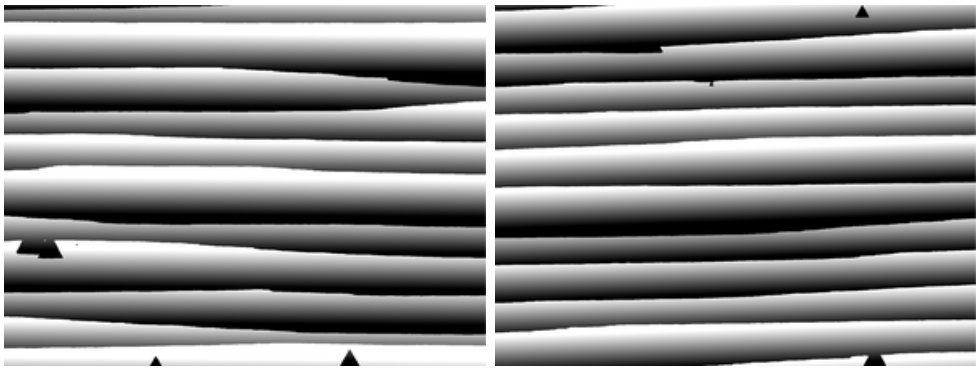
With the modified SOS model, the simulation model can now be used to directly simulate the surface morphology of various surface configurations and provide a direct comparison to our experimental results.

Figure 4.4 to Figure 4.6 show the steady state morphologies for various miscut configurations obtained with the simulations. The simulation uses the following parameters: $k_{terr} = 1 \times 10^{-10}$, $k_{dopant} = 1 \times 10^{-7}$, $k_{dislocation} = 10$, $\rho_{dopant} = 1 \times 10^{-4}$,$\rho_{dislocation} = 10^{-7}$. For surfaces with large miscut angles such as 0.12° (figure 4.4), the pit effects are suppressed, thus the surface morphology is quite uniform and a small area of simulation can give good agreement with experimental morphologies. For smaller miscut angles such as 0.09° (figure 4.5), the pit effects become more pronounced and produce more irregular step distributions. A 2×1.5 μm window is actually not enough to reflect the over all surface morphology. Thus, with a small simulation area, it may never produce a satisfactory agreement between the simulation and experiments. In figure 4.5, although the simulation with the selected parameters produced morphologies

that agree well with selected experimental morphologies, they do not represent the overall morphology and additional parameter tuning is required to give better agreement. For surfaces with very small miscut angles, the size of the simulation window becomes a more critical limit. Figure 4.6 simulates some of the properties of the surface morphology that was observed in experiments on samples with extremely small miscut angles. However, satisfactory agreement between the actual experimental surface morphology and that from simulations is not possible with the current available computing resource.

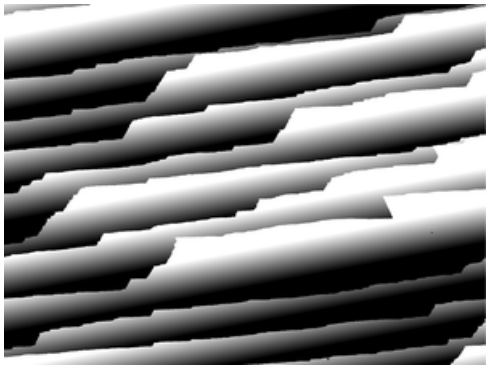
The overall relationship between the miscut and steady state morphologies is apparent. For large miscut, the pits are suppressed. As the miscut angle gets smaller, there are more pit effects such as step edge roughness and the overall fragmentation of the steps and terraces increases. For the same miscut angle, the miscut orientation also plays a role. As the miscut orientation deviates from $< 11\bar{2} >$, the steps flow faster, which suppresses the pitting effects.

One should note that this set of results is obtained by optimizing the parameters to match our experimental results (particularly our experiments on samples with 0.12° miscut) best. For a different wafer, the parameters have to be tuned and the results are likely to be different. As an example, for a wafer with more crystal defects, more stacking pits may be produced. For wafers with higher doping, the surface will have a higher pitting density and more pitting effects may show up for surfaces with relatively large miscuts.

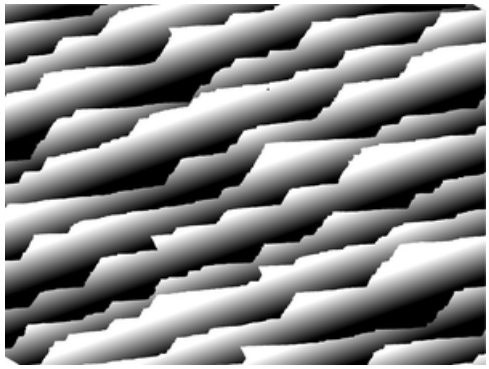


(a) sim: 0° off $< 11\bar{2} >$

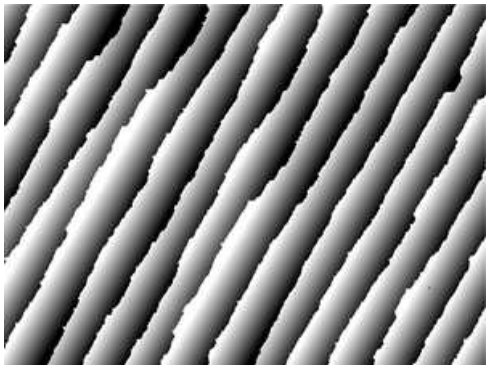
(b) sim: 2° off $< 11\bar{2} >$



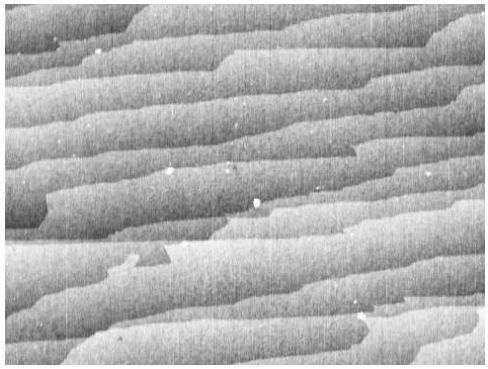
(c) sim: 10° off $< 11\bar{2} >$



(d) sim: 20° off $< 11\bar{2} >$

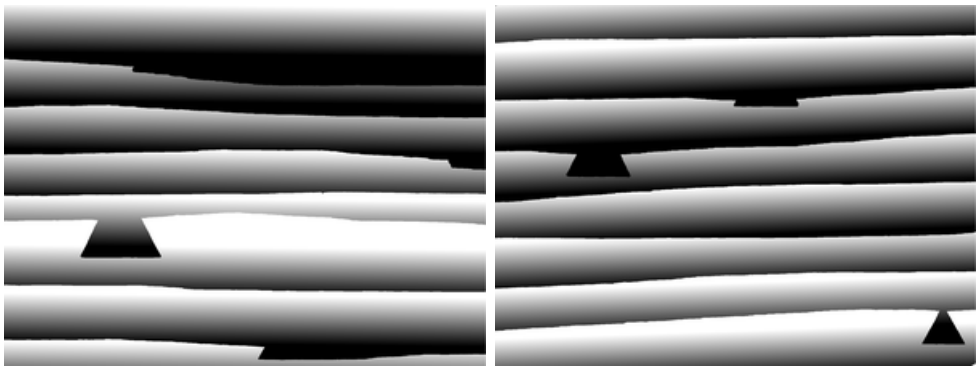


(e) sim: 60° off $< 11\bar{2} >$



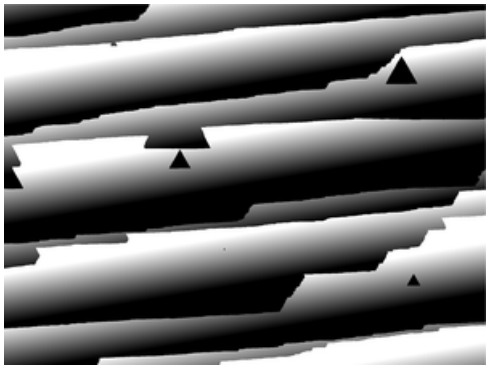
(f) exp: 10° off $< 11\bar{2} >$

Figure 4.4. (a)-(e) Simulated steady state morphology for surfaces with 0.12° miscut at various misorientations, (f) Experimental results with surface miscut angle 0.12° miscut and 10° off $< 11\bar{2} >$. Image size is 2×1.5 μm.

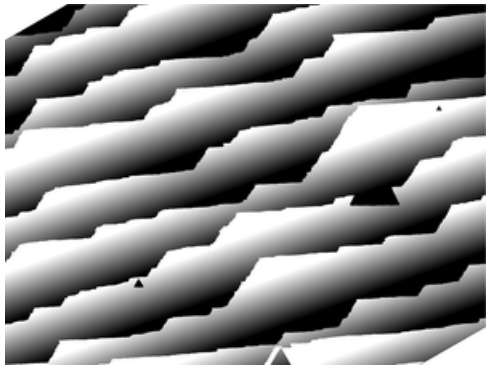


(a) sim: 0° off $< 11\bar{2} >$

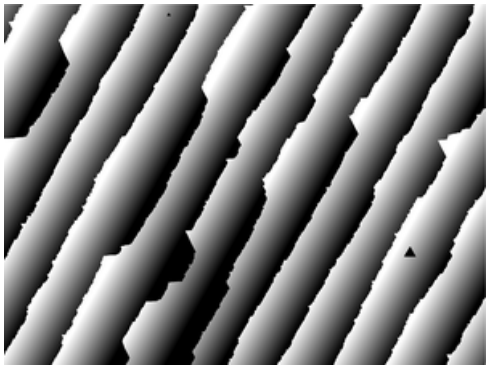
(b) sim: 2° off $< 11\bar{2} >$



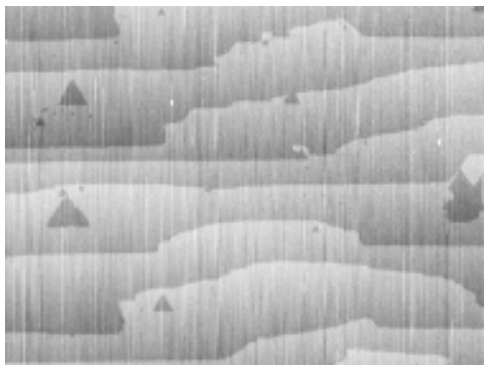
(c) sim: 10° off $< 11\bar{2} >$



(d) sim: 20° off $< 11\bar{2} >$

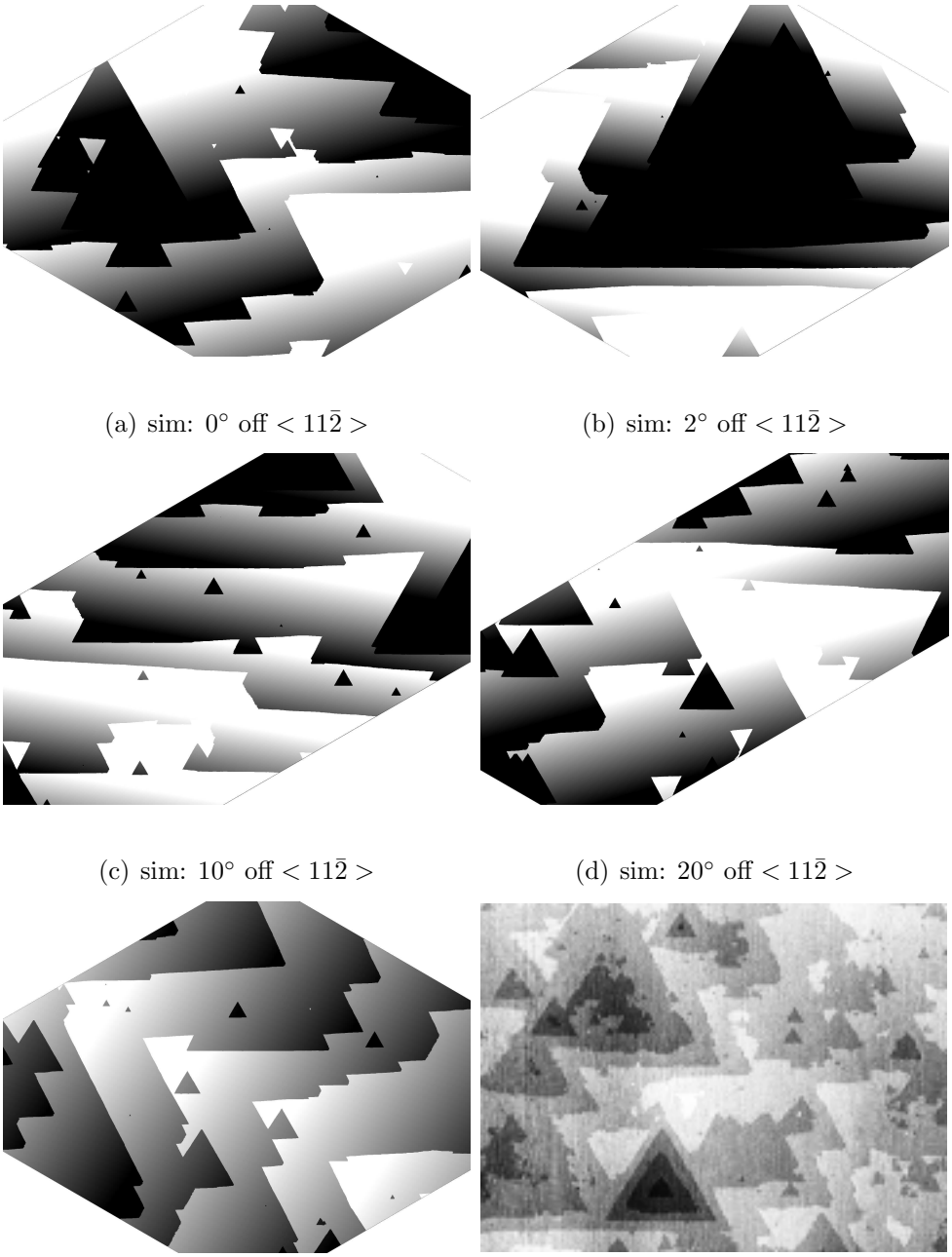


(e) sim: 60° off $< 11\bar{2} >$



(f) exp: 2° off $< 11\bar{2} >$

Figure 4.5. (a)-(e) Simulated steady state morphology for surfaces with 0.09° miscut at various misorientation, (f) Experimental results with surface miscut angle 0.09° miscut and 2° off $< 11\bar{2} >$. Image size is 2×1.5 μm.



(a) sim: 0° off $< 11\bar{2} >$

(b) sim: 2° off $< 11\bar{2} >$

(c) sim: 10° off $< 11\bar{2} >$

(d) sim: 20° off $< 11\bar{2} >$

(e) sim: 60° off $< 11\bar{2} >$

(f) exp: 20° off $< 11\bar{2} >$

Figure 4.6. (a)-(e) Simulated steady state morphology for surfaces with 0.02° miscut at various misorientations, (f) Experimental results with surface miscut angle 0.02° miscut and 20° off $< 11\bar{2} >$. Image size is 4×3 μm.

4.4 the morphology dependence on initial etch time

Another important result in our experiments is modeling the time dependent surface morphologies during etching. During the processing of a surface, the question arises: what length of etching time is necessary for the surface to recover from its initial roughness to the steady state morphology. Our experiments indicate that generally longer times are required for smaller miscut angles. More precise information is difficult to obtain experimentally as it not only requires a variety of samples with different miscut configuration, but also requires quite labor intensive experiments for each sample. Furthermore, for surfaces with large miscut angles, it is quite difficult to measure the time dependent initial morphology as the surfaces etch quite fast. In this section, we will try to use kinetic Monte-Carlo simulations to study the time-dependent initial morphologies. Although the results obtained through simulations are not strictly rigorous, it will help us gain more understanding of the initial development of the surface.

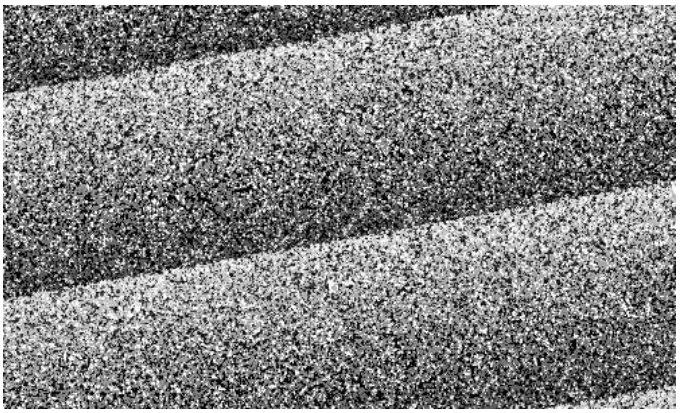
To simulate the initial etching, the most difficult challenge is to model the initial surface. The initial surfaces following the HF etching and removal of the silicon oxide layer are atomically rough. Although the roughness of the surface is often measured to be below 1 nm or within a few monolayers, there are no recognizable steps or terraces. To generate such a random surface, while

maintaining the overall miscut configuration and the SOS model integrity is not trivial.

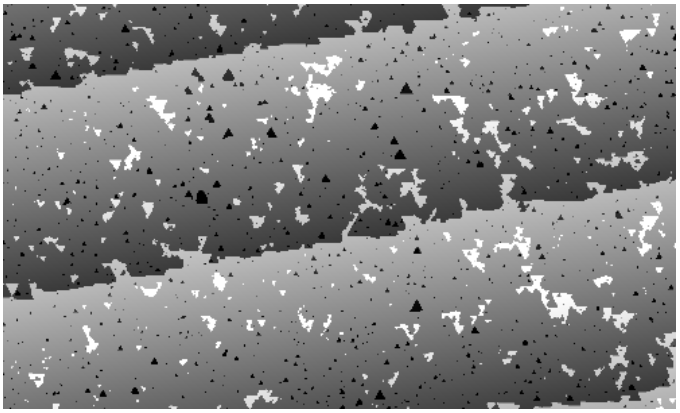
In the simplest approach, the surface can be roughened by etching the surface with the same etch rates for various classes of atom sites. This etching provides a completely random etching effect. The results of such treatment are shown in figure 4.7. Although the random etching produced an atomically rough surface, the overall terraces are still apparent, and it is very clear where the underlying steps are located as they remain straight. The result does not represent the actual initial surface. Short initial etching quickly removes the roughness, and the results do not agree with the observations in our experiments.

In order to further obfuscate the initial step edge information, the surfaces were first etched with $k_{mono} = k_{di} = k_{kink} = 10^{-4}$, and $k_{terr} = 10^{-12}$ while the other parameters remained unchanged. After 4 layers of etching, both the surface steps and the terraces are no longer apparent (figure 4.8b). To further increase the surface noise, the surface was continuously etched with an increasing terrace etch rate. For each stage, the terrace site etch rate is increased by 1 order of magnitude, until the last step, when the surface was etched with all site-specific etch rates equal (10^{-4}). The final surface is shown in figure 4.8d, which matches the initial surface in experiments much better than figure 4.7.

Figure 4.9 shows simulated results for the initial etching of a Si(111) surface with 0.12° miscut and 10° misorientation off the $< 11\bar{2} >$ direction. The relative etch time for each snapshot is shown in the figure. To compare with our

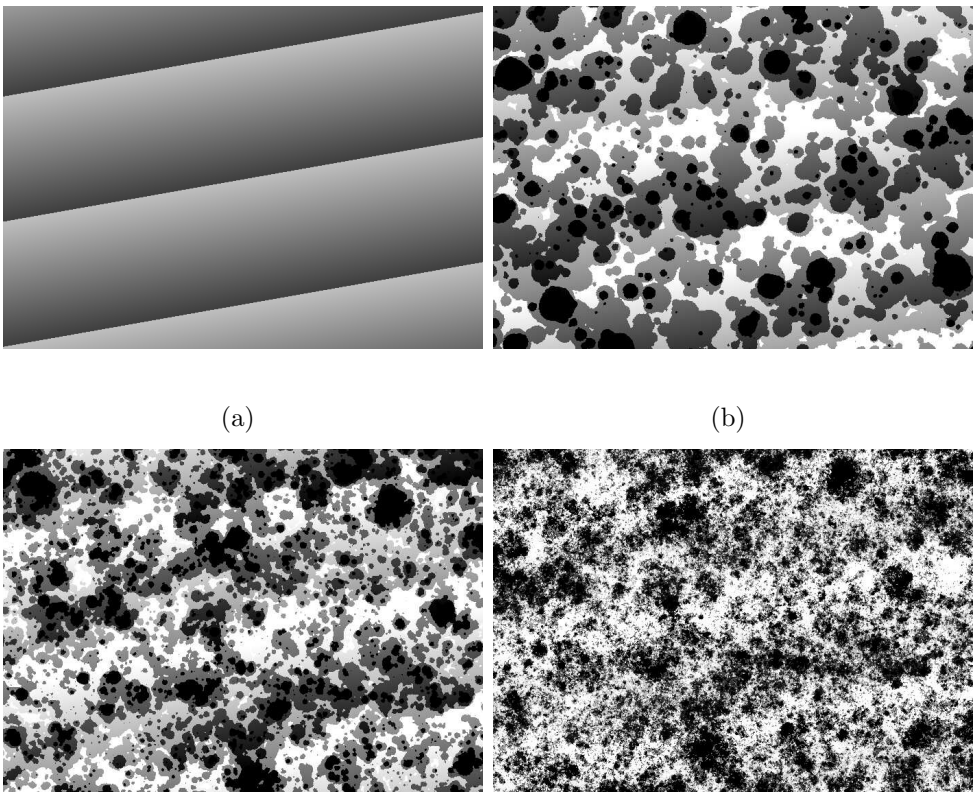


(a)



(b)

Figure 4.7. (a) Initial rough surface simulated by randomly etching an otherwise perfect surface. (b) The surface quickly becomes a smooth surface with short etching time.



(a) (b)

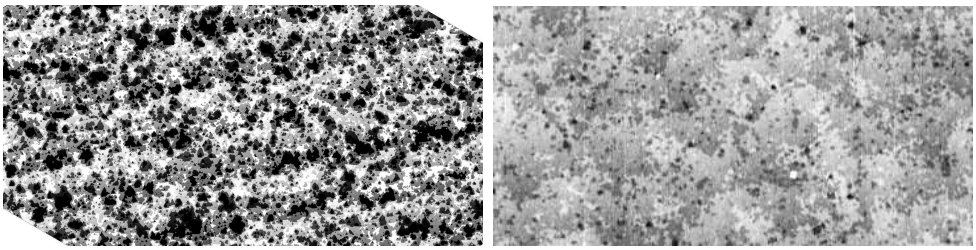
(c) (d)

Figure 4.8. Preparation of the initial surface. To simulate the initial surface before etching, we start with a perfect surface (a), then etched with $k_{mono} = k_{kink} = k_{di} = k_{point} = 10^{-5}$, and $k_{terr} = 10^{-19}$(b); then continuously etched with increasing k_{terr}(c); until $k_{terr} = 10^{-5}$(d).

experiments, the simulation etch times were converted to real units of time using equation 3.10.

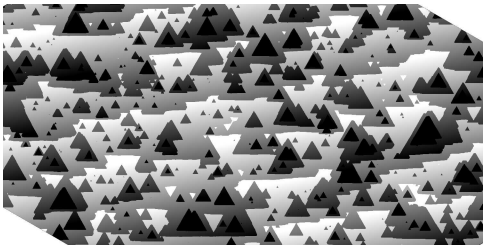
The initial etching is quite fast as the initially rough surface contains large quantities of high reactivity sites. After 3 monolayers are etched away, the surface is already dominated by single atomic steps and single atomic terraces. The overall surface is still dominated by small etch pits. After 6 monolayers of etching, long range terraces become apparent and after 9 monolayers of etching, the surface reaches steady state morphology. The results again agree well with our experiments.

Figure 4.10 shows the simulated results of initial etching from a Si(111) surface with 0.02° miscut. Due to computational resource limits, the simulation was restricted to an area with only two terraces. Due to the constraints of the simulation area, to show the effects of multiple pit formation, the etch rates of the dopant and defect sites were slightly increased, to give better agreement between our experiments and simulations.

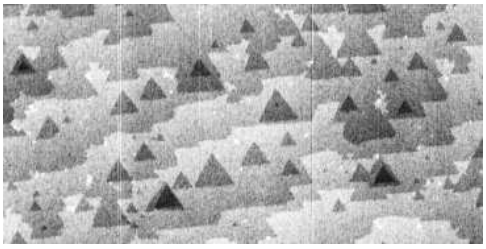


(a) sim: t=435 (2 sec.)

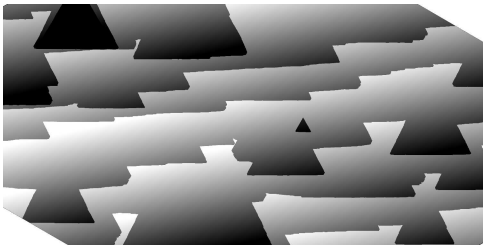
(b) exp: t=1 min.



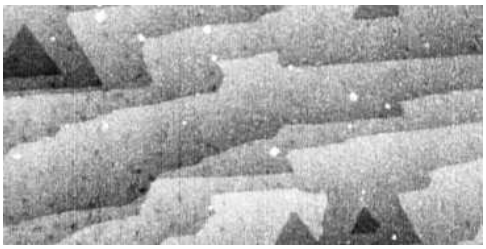
(c) sim: t=13606 (1.0 min.)



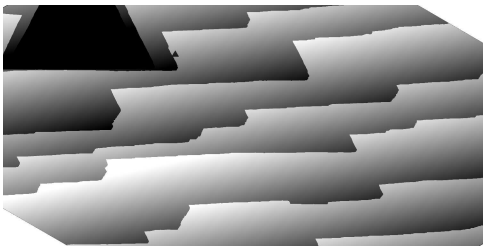
(d) exp: t=2 min.



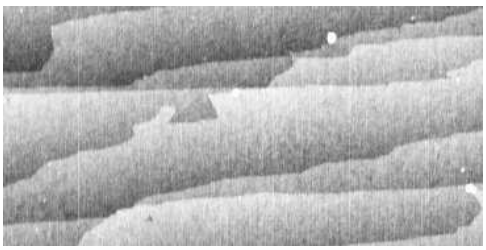
(e) sim: t=43094 (3.2 min.)



(f) exp: t=4 min.

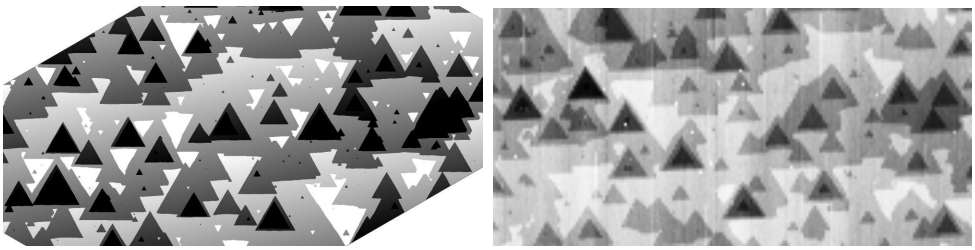


(g) sim: t=79033 (6.0 min.)



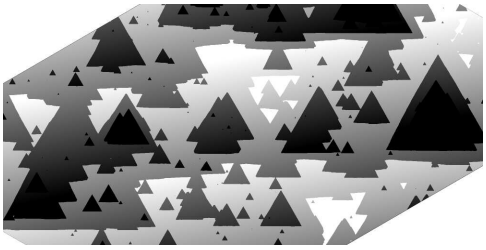
(h) exp: t=6 min.

Figure 4.9. Simulated surface morphology and experimental results during initial etching on surfaces with 0.12° miscut . Etch Rates: $k_{defect} = 10$, $k_{dopant} = 10^{-6}$, defect concentration 10^{-7}, and dopant concentration 10^{-4}.Image sizes are 1500 × 750 nm.

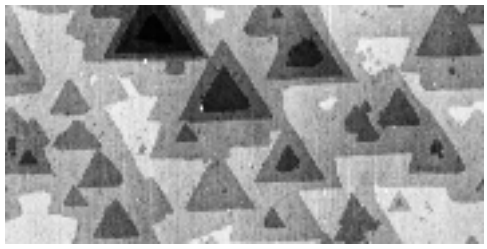


(a) sim: t=61351 (4.6 min.)

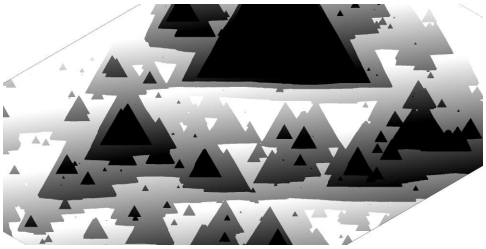
(b) exp: t=5 min.



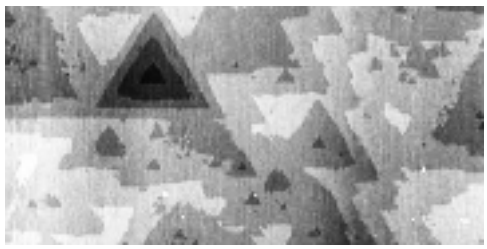
(c) sim: t=198833 (14.9 min.)



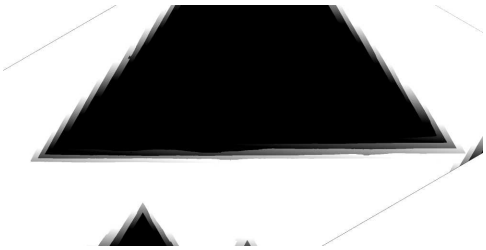
(d) exp: t=15 min.



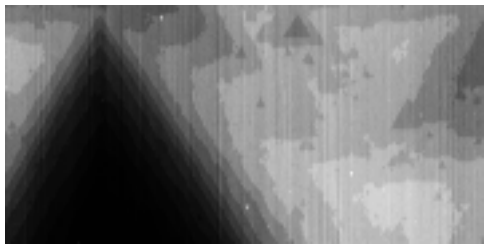
(e) sim: t=601786 (45.2 min.)



(f) exp: t=45 min.



(g) sim: t=1391540 (1.7 hr.)



(h) exp: t=3 hr.

Figure 4.10. Simulated surface morphology and experimental results during initial etching on surfaces with miscut 0.02°. Etch Rates: $k_{defect} = 10$, $k_{dopant} = 10^{-5}$, defect concentration 5×10^{-6}, and dopant concentration 10^{-4}. Image sizes are 4×2 μm.

Chapter 5

STM Based Surface Modification of Hydrogen-Terminated Si(111) Surfaces

5.1 Introduction

The current semiconductor industry is largely based on continuing advances in photolithographic technology. State of the art manufacturing devices use 193-nm lasers and produce chips with 130-nm feature linewidth. The current complementary metal oxide semiconductor (CMOS) technology is likely to function even below 50-nm linewidths. However, to produce chips beyond this linewidths on novel devices and structures, the development of novel lithography technologies is required. Scanning tunneling microscopy (STM) and atomic force microscopy (AFM) are among the non-conventional nanofabrication tools in modern laboratories [60]. The STM probe provides an intense electron current density through tunneling with very low energies, which is capable of changing the chemical structures of the hydrogen terminated silicon surfaces either in air

or vacuum [61, 18]. Due to the extremely small radius of the scanning probe (tip) and the small tip-surface separation, the tunneling electrons emitted from the tip are well confined and capable of producing patterns with linewidths below 10 nm [18, 12].

The nano-patterning of H-terminated silicon surfaces can successfully produce positive results both in an ambient environment [16] and ultra high vacuum (UHV) [18]. Although both processes share similar attributes such as the relation between the writing efficiency and bias voltage and electron dosage, the underlying mechanisms are likely to be different. In an ambient environment, due to the abundance of water molecules, the process directly produces an oxide pattern, which can be directly used as an etch mask to further transfer the patterns. The linewidths of the pattern created in air are typically from 20 nm to 100 nm. In ultrahigh vacuum, due to the highly controlled environment, it is commonly believed the patterns are produced by directly desorbing hydrogen termination by the tunneling electron current. However, in a common vacuum system without sufficient baking, water vapor can be a dominant gas species within the partial pressures of the various residual gases, and water may still play an important role during the patterning process. In a UHV environment, the role of water is largely suppressed, and the patterning process typically achieves smaller linewidths than that in air. Patterns with linewidths of 1 nm have been reported using a hydrogen terminated Si(100) surface as the substrate [38].

The patterning process in an ambient environment has been studied quite

extensively. One of the limits of the ambient process is due to the uncontrolled environment. The originally atomically ordered surfaces are easily contaminated and it is difficult perform STM imaging with atomic resolution, thus preventing us from achieving atom-based patterning. The UHV patterning process is capable of fabricating much narrower linewidths. With an atomically ordered surface lattice, it is possible to fabricate patterns with atomic accuracy. We are currently focused on the study of the UHV patterning process on the wet chemically prepared H-terminated Si (111) surfaces.

The patterns immediately following creation are imageable with the STM that was used to create the pattern. Before being fully useful as reference standards, the patterns have to be transfered into a more accessible pattern with significant relief. For example, the patterns have to be locatable and have enough contrast for AFM or SEM to access. This has been the bottleneck preventing the application of the UHV patterning process. A common high-resolution UHV STM doesn't have a macro motion and macro vision system, and usually has a very limited scanning range (a few micrometers). Thus, to find the patterns created with a UHV STM using a second tool such as AFM or SEM is like finding a needle in a haystack. This makes the study of pattern transfer techniques next to impossible. To address this problem, we have studied the pattering process in a low vacuum system in which an STM is equipped with a camera vision system which enables us to locate the patterns to within 50 micrometer position accuracy. We have demonstrated a successful pattern

transfer. Currently a new UHV STM system is being built with both a micro motion system — capable of movement at 1 nm precision in 1 cm range— and a macro vision system with 2 micrometer resolution. The new system should be able to link the UHV patterning process with more general instruments and bring new exciting applications.

5.2 Patterning process in ambient environments — scanned probe oxidation

The scanning probe microscope(SPM) induced modification of hydrogen-terminated Si surfaces in an ambient environment is believed to follow a mechanism called scanned probe oxidation. It was first reported by Dagata, a coworker at NIST [61]. In this study, a hydrogen-terminated Si(111) surface was scanned in air by a positively biased tip to generate surface oxide features. The created patterns achieved a size of 100 nm. The experiments were followed by more STM [62, 63, 64] and AFM studies [65, 66, 67]. Creation of patterns with linewidth as narrow as 10 nm has been reported [67] and nanoelectronic devices have been fabricated using this process [68, 17, 69].

There have been many discussions on the mechanisms of scanned probe oxidation [70, 71, 72]. A commonly accepted mechanism is the following. Applying a voltage between a conductive SPM tip and the hydrogen terminated

sample surface with the surface being positively biased results in the formation of a highly nonuniform electric field. The E field attracts a stable water meniscus to the sample junction. After initial oxidation, the surface become hydrophilic, which in turn enhances the formation of water meniscus. The oxyanions, O^- and OH^-, produced electrochemically at the sample surface by the SPM tip are driven by the electric field through the growing oxide and react with holes, h^+ and H^+, at the silicon-oxide interface to produce a silicon oxide [71]. The thickness of the oxide formed by this process is found to be a function of both bias voltage and time. Garcia has reported a relation

$$dh/dt \approx exp(-h/L_c), \tag{5.1}$$

where h is the oxide thickness, $L_c(8\text{V}) = 0.5\text{nm}$, and $L_c(12\text{V}) = 0.6\text{nm}$ [72].

5.3 Patterning process in ultrahigh vacuum — electron stimulated desorption

Although atomically flat silicon (111) surfaces have been routinely prepared under an ambient environment, there have been no substantiated reports that the wet chemically prepared surfaces can be imaged in air with atomic resolution. This could be due to the initial oxidation, which can start as soon as 15 minutes after preparation [73]. During the initial oxidation, although the Si–H bonds may remain intact, oxygen can still be Incorporated into the Si–Si backbonds [31, 74].

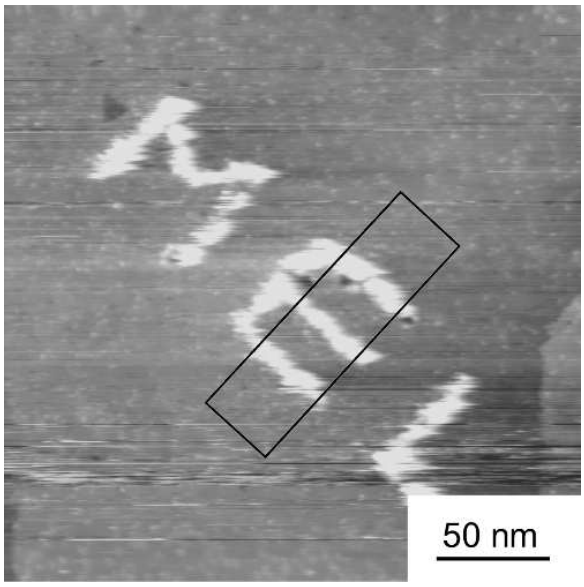
During the STM imaging, the field between the tip and the sample attracts a water meniscus, and it likely to cause a "defocussing" effect [70] and reduces the lateral resolution of the STM. To achieve an atom-based patterning process, UHV environment is desired.

In 1990, Higashi et al. reported that the hydrogen termination can be desorbed in UHV by STM with a positive bias and creating an unsaturated silicon region as small as 40 angstroms [13]. The process was described as electron stimulated desorption (ESD). Later, Lyding et al. evolved this technique into a UHV nanoscale STM patterning process on hydrogen-terminated Si(100)-2×1 surfaces [38, 18, 75]. They demonstrated creating patterns with linewidths as narrow as 1 nm superimposed on atomically ordered surfaces [38]. In their experiments, the hydrogen-terminated Si(100) surface was prepared using a conventional high temperature annealing process followed by leaking atomic hydrogen to terminate the surface. A similar technique has been used to obtain hydrogen-terminated Si(111):7×7 surfaces [76]. As there is no effective way to pre-mark the surface which will survive the annealing process, wet chemically prepared hydrogen-terminated Si(111) surface is highly desirable. Schwartzkopff et al. extended this process to wet chemically prepared H-terminated Si(111) surfaces in high vacuum and reported producing line patterns with linewidths from 5nm to 10 nm [12]. However, their experiments were performed in a high vacuum environment, and it is a question whether the process had the same mechanism as the process in UHV.

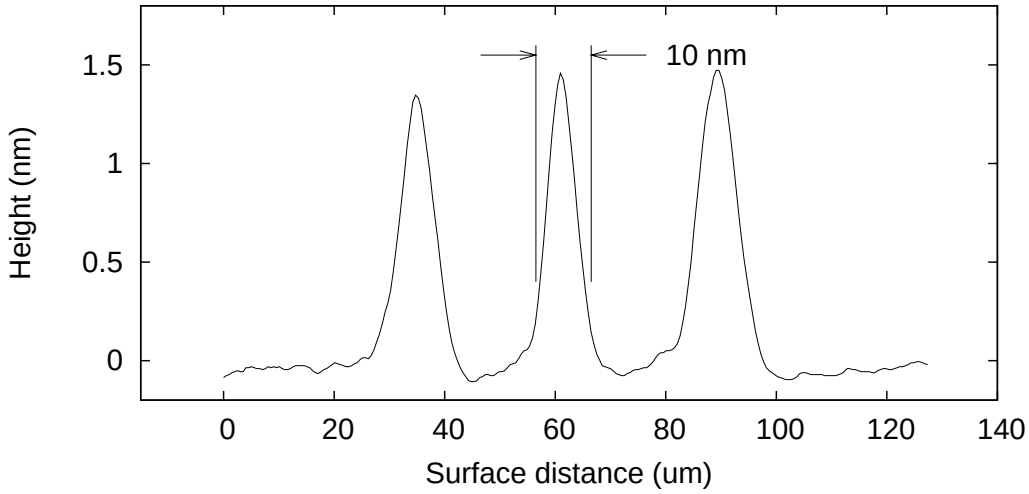
Our experiments are performed in a UHV system with a background pressure of 8×10^{-8} Pa. The hydrogen terminated Si(111) surfaces were prepared with the procedures described in chapter 2. Boron doped (3-6 Ωcm) Si(111) samples with miscut 0.12° were used. All patterning process and the followed imaging are performed with a high resolution UHV STM. AC etched W tips with typical radius of 15 nm [77] are used.

Figure 5.1 shows an STM image of the result of a successful patterning process. The pattern was created by scanning the tip at a speed of 20 nm/sec, and a sample positive bias voltage of 6 V. During the writing, the feedback was kept on maintaining a constant tunneling current of 1 nA. The STM image was measured immediate after the patterning process at a bias voltage of -1.7 V.

The patterns show up under STM imaging as an elevation in height around 1.5 nm. This can be explained by the analysis of the band structure of the hydrogen-terminated surface. For hydrogen-terminated silicon surface, there are no states above the Fermi level and the theoretical slab calculations by Schluter and Cohen [78] predict the position of the Si-H bonding and anti-bonding band at around 3 eV below the Fermi level for the Γ point of the Brillouin zone. After the hydrogen desorption, the silicon dangling-bond states form and results in a slight downward band-bending characteristic in the near-surface region. These states offer a more efficient electron transfer between the STM probe and sample surface. To maintain the constant current mode, the feedback retracts the piezo and gives the small elevation in height over the patterned area.



(a)



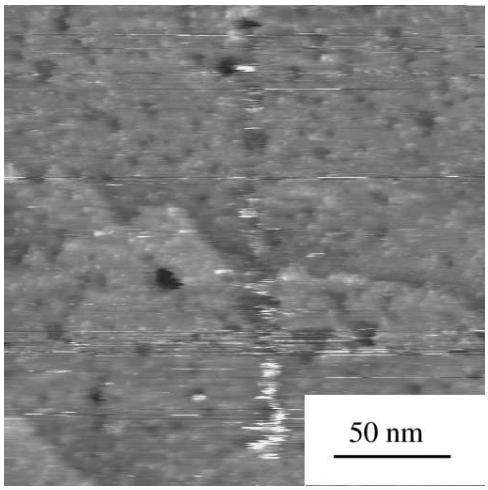
(b)

Figure 5.1. STM images of a nano pattern created in UHV. (a) A topography view. (b) The cross-section view of the box area marked in (a). ("MEL" stands for "Manufacture Engineering Lab").

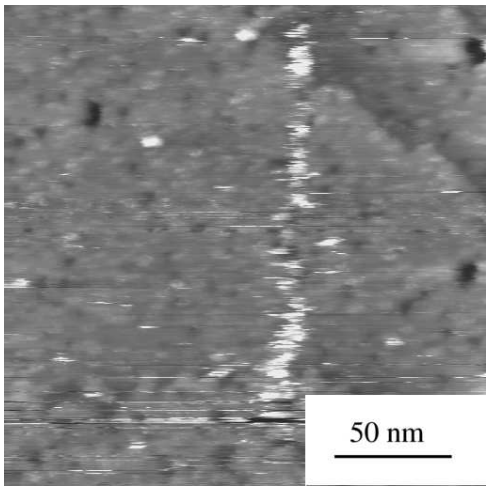
The patterning process was observed to be affected by both the amplitude of the bias voltage and the time duration the probe stays above the patterned sites. Figure 5.2 shows images of two lines patterned at 4 V and 5 V respectively, along with a pattern created at 6 V. All these patterns are written with a tip scanning speed of 15 nm/sec. The patterned lines are more continuous with the increased bias voltage. Both the linewidths and apparent height of the pattern do not display an pronounced change. The bias voltage changes the energy of the electron emitted from the tip and the above data indicates that increasing the electron energy increases the efficiency of the hydrogen desorption.

Figure 5.3 shows three patterns that were written at 6 V bias voltage, 0.8 nA tunneling current, but with a tip scan speed of 20 nm/sec 40 nm/sec and 100 nm/sec. The pattern written at slow scan speed results in a continuous pattern, while moving faster results in broken pattern lines. The attempt at 100 nm/sec didn't result in a recognizable pattern at all. Neither the linewidth nor the apparent height of the pattern lines in the former two images are affected by the tip moving speed.

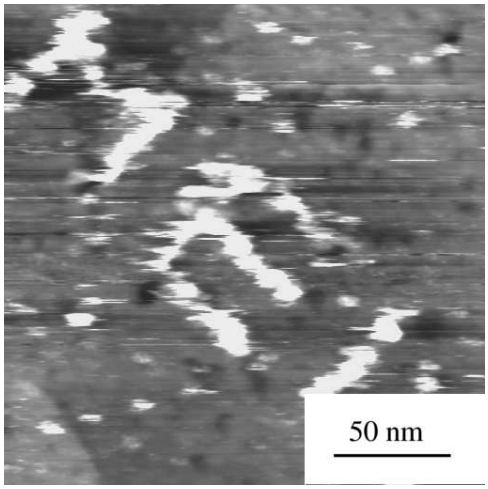
If we assume a complete hydrogen desorption of the patterned area, the desorption yield for the structure of figure 5.1 is around 2.5^{-7} H atoms per electron. This value is larger than that reported by Becker [13] (3^{-8}) and Schwartzkopff [12] (1.4^{-8}). However, our experiments indicate the the pattern linewidth exhibit no strong relation with either bias voltage or electron dosage, but rather strongly affected by tip conditions, which may vary the distribution of



(a) 4.5 V

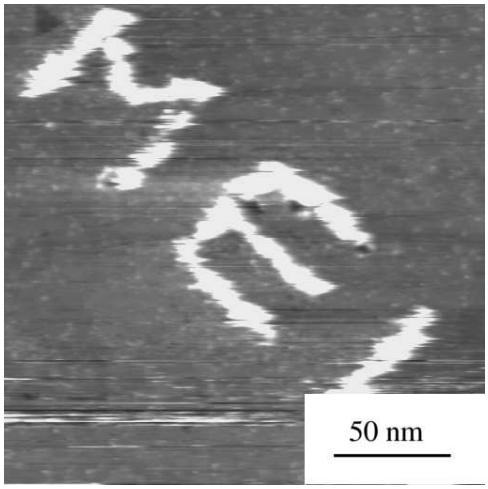


(b) 5.0 V

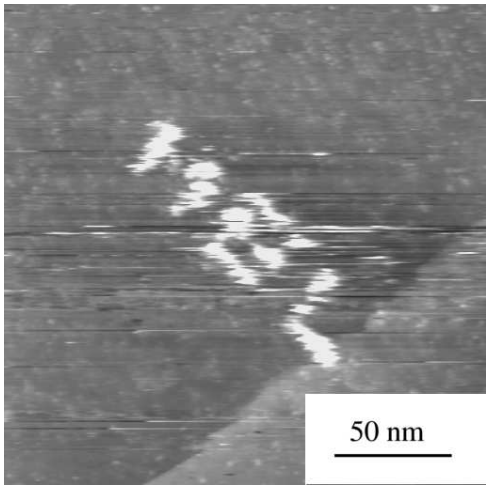


(c) 6.0 V

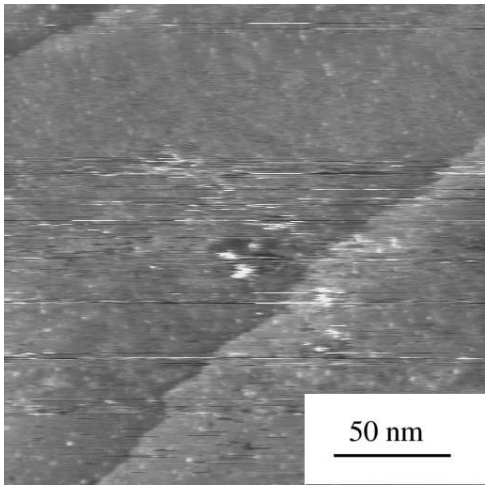
Figure 5.2. STM image of patterns created with a scanning speed of 15 nm/sec, 1 nA tunneling current, and with bias voltages of 4.5 V 5.0 V and 6.0 V.



(a) 20 nm/sec



(b) 40 nm/sec



(c) 100 nm/sec

Figure 5.3. STM images of patterns created with 6 V bias voltage, 0.8 nA tunneling current, and at various tip scanning speeds.

electron dosage across the patterned area. Thus, the simple calculation of electron yield may not be a reliable method to compare with.

Two mechanisms have been proposed for the UHV patterning process: (1) multiple vibrational excitation of the Si–H bond induced by inelastic electron tunneling and (2) direct excitation of the Si–H bond by the field-emitted electrons [38]. The energy of the electrons emitted from the STM probe after penetrating an effective tunneling barrier (estimated to be about 3 eV) is typically less than 3 eV. The bond strength of the Si–H is 3.5–3.8 eV [79], and the excitation from Si–H bonding states to anti-bonding states requires about 5.8 eV. Hence, the multiple-excitation mechanism is favored. During the patterning process, the increase of bias voltage effectively increases the energy of the electrons emitted from the tip, thus increasing the quantum yield of the hydrogen desorption process. With an increase in the tip speed, fewer electrons hit the surface per unit area, thus decreasing the overall pattern strength.

5.4 Experiments in low-vacuum environments

One of the keys to transferring a pattern created by a UHV STM is to locate the STM probe position during the UHV patterning process. One solution is to pre-mark the sample surfaces and, using a optical microscope with a long focal length, to locate the tip relative to the mark with an uncertainty of a few microns. This is not very difficult to implement in an ambient environment,

where there is no reason for concern about UHV compatibility. The task of integrating a high-magnification optical tool into a UHV STM is formidable, but feasible in principle. To demonstrate the feasibility, we experimented with the patterning process on hydrogen-terminated Si (111) in a low-vacuum facility in which there was an STM system integrated with a camera and 2-axis high-resolution interferometers. Without being concerned about maintaining a high vacuum, we successfully relocated the pattern written in vacuum by using an AFM operated in air. Having the capability of relocating the pattern also enabled us to experiment with pattern-transfer techniques. As a first attempt, we experimented with reactive-ion etching (RIE) and successfully increased the pattern feature height by a factor of 4.

Since the vacuum condition was much less stringent, protection of the surface after chemical etching was less critical than in the UHV experiments. After the samples were prepared, they typically had an exposure to air of about an hour before they were loaded into the vacuum chamber. During the experiments, all the ion gauges and ion pumps were intentionally turned off, as severe surface roughening has been observed with the ion pump running. The vacuum chamber was pumped down by a turbo pump and had a background pressure of 10×10^{-6} Pa. During STM operation, the turbo pump was turned off to reduce the vibration, and the vacuum level would creep back up to 0.1 Pa (dominated by water vapor).

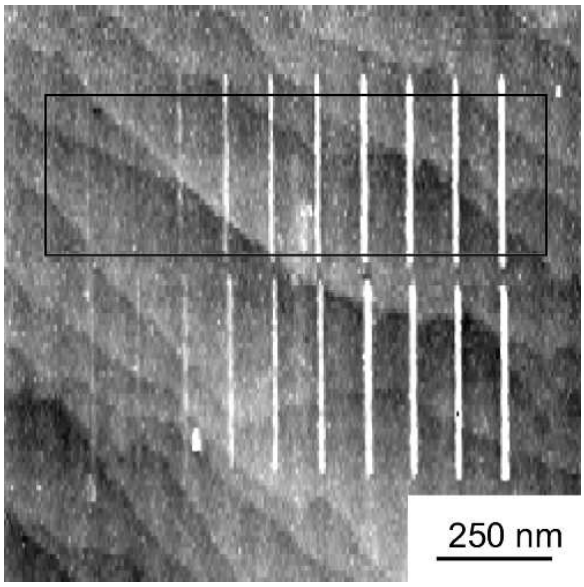
Figure 5.4 shows one of the patterning results under low vacuum. A bias

voltage ranging from 3.5 volt to 4.5 volt at equal steps of 0.1 volt was used to create the line segments. Its averaged cross-section is shown in Figure 5.4(b). For bias voltages less than 3.9 volts, the feature height was observed to be proportional to the bias voltage, a relationship that was less pronounced when the bias voltage rose above 3.9 volts. The widths of features in all these line patterns are about 20 nm.

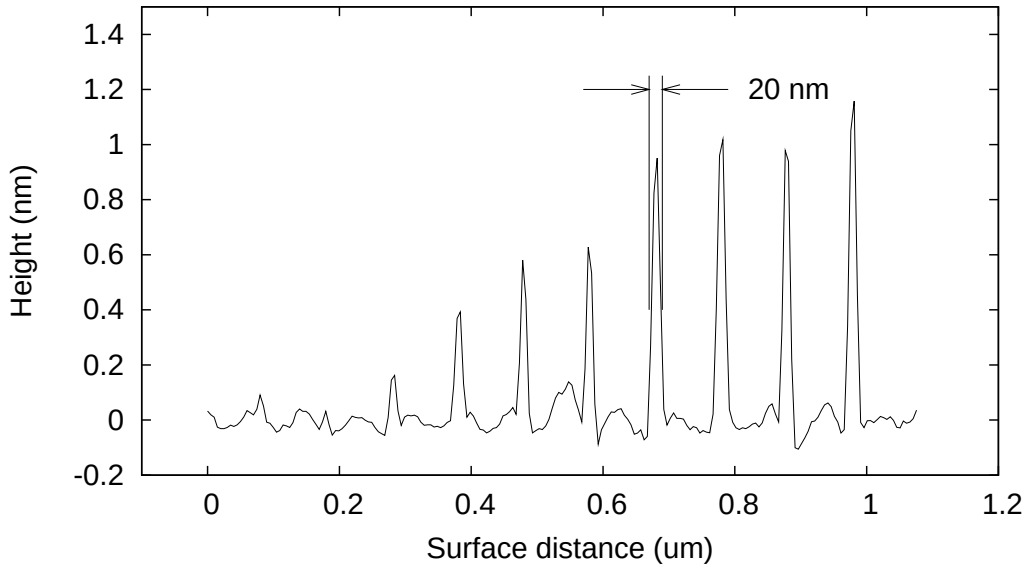
Unlike the UHV experiments, the patterned lines were mostly continuous at all the bias voltages. Changes in the bias voltage difference resulted in variation of the apparent line height of the patterns. This line height was less than that which we observed in UHV. These results suggest that the patterned area is likely to have been covered with an oxide layer under the low-vacuum conditions.

The exact mechanism governing the patterning process is not clear at this point, but the following mechanism has been proposed: The surface after wet-chemical processing is hydrophobic. It has been reported by Takahagi et al. to be water free [80]. Although the water molecules is the dominant species among the low-vacuum residue gases, the partial pressure of water in vacuum (below 0.1 Pa) was significantly lower than it would have been in an ambient environment (typically 100 Pa), and it was thus more difficult to form a water meniscus between the tip and the sample surface.

We propose the following complementary mechanism: Because of the long duration of the exposure to air and the low-vacuum conditions, the oxygen started to be incorporated into the Si–Si bonds at the back of the surface silicon



(a)



(b)

Figure 5.4. STM image of a set of line segments created in a low-vacuum environment. The image was imaged immediately after the patterning process. From left to right, the lines were written as the bias voltage was increased from 3.5 V to 4.4 V in steps of 0.1 V. Image courtesy of George Witzgall.

hydride. During the patterning, the Si–H bonds were vibrationally excited by the electrons emitted from the tip, facilitating a transition from Si–O–Si–H to Si–O–Si–OH because of the residue of water. At that point, the surface became hydrophilic and attracted more water molecules, even to the point of forming a film of water and further oxidizing the surface.

5.5 Pattern transfer of the nano-structure fabricated in vacuum

We had little difficulty using an air-operated AFM to locate the pattern created during the low-vacuum patterning process. The oxide patterns were very stable in air, and repetition of the scanning a few days later revealed no change in the pattern dimensions. The patterns had a height contrast of about 4 nm as measured by AFM.

Silicon oxide provides a hard etch mask, which makes it possible to further etch the sample to increase the aspect ratio of the pattern structures. However, in our case, the oxide that covers the patterned area was much thinner and narrower than in typical applications, which presented a challenge to the pattern-transfer process.

Hydroxide etching of silicon is commonly used to transfer patterns onto a silicon substrate [10]. A hydroxide-based solution etches the silicon by first

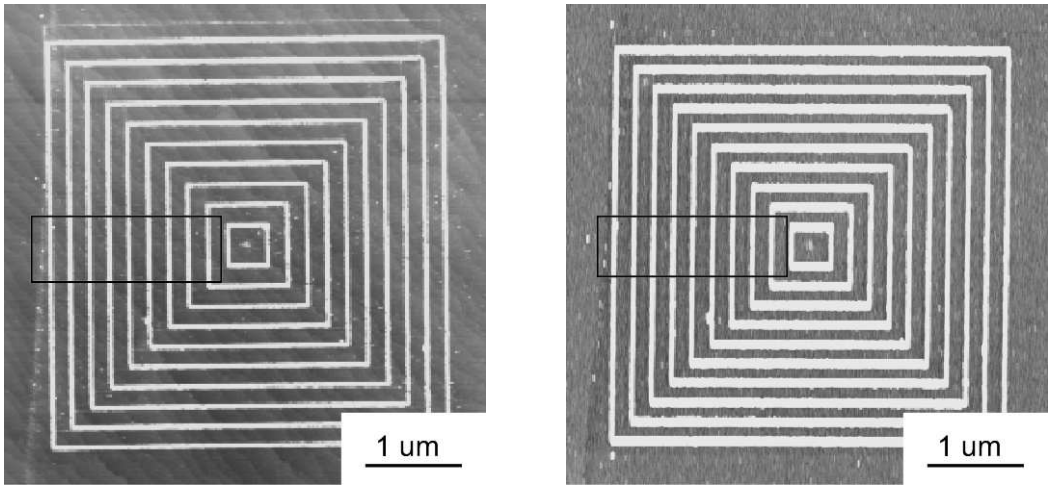
oxidizing it to form a silicate, and then the silicate reacts with hydroxy ions to form a water-soluble complex. Typical hydroxide-based etching solutions include KOH, NaOH, NH_4OH, and tetramethyl ammonium hydroxide (TMAH). For our first attempt at etching, we experimented with using 0.5 M of KOH buffered with isopropanol in a ratio of 1:1 by volume. After only 1–2 seconds of etching, the pattern was destroyed, with only a few traces remaining. That result was partially expected on account of the strong anisotropy in KOH etching of silicon, which is approximately 1:600:400 for Si (111):(110):(100). Although such a strong aniostropy works perfectly for pattern transfer on Si (110) substrates, it fails on Si (111) substrates.

Further experiments with TMAH etching are planned. TMAH etching of silicon offers a milder anisotropy, which is around 1:10 to 1:35 for silicon (111):(100). Another advantage is that the oxide mask works better with TMAH etching, as it does not significantly etch the oxide.

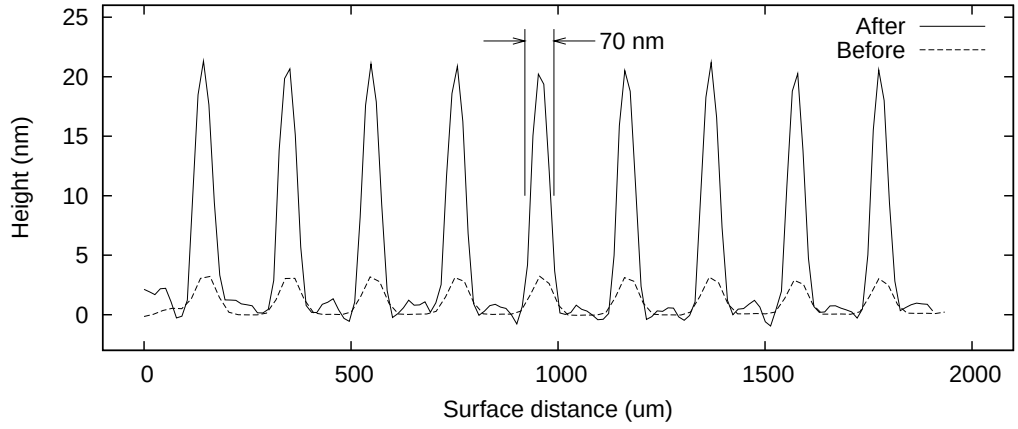
Another alternative is to use reactive-ion etching (RIE). During that process, a plasma is created by applying an electric field of a certain magnitude to a gas. The RF energy at 13.56 MHz provides the source that causes a neutral gas to decompose and ionize in the reactor chamber, which in turn brings about the formation of ions, electrons, and free radicals. For SF_6, a gas that is commonly used for silicon etching, F radicals are formed in the plasma. These F radicals are attracted to the silicon substrate along with the ions and cause the surface to be etched. Typically, RIE etching consists of a combination of etching via ion

sputtering and chemical etching by the F radicals. This process is also known as accelerated ion-assisted etching, and it can take place under nearly isotropic conditions.

Figure 5.5(a) shows one of the patterns imaged by AFM. For this particular pattern, the linewidth is about 70 nm, which is consistent with the results of the low-vacuum STM measurements. It is very difficult to initially locate the pattern with an AFM due to the uncertainty of the position measurement in the low vacuum chamber. The pattern was finally located with the AFM after almost a week of searching. The same sample was then etched in an RIE facility. Figure 5.5(b) shows the resulting pattern obtained by application of RIE with SF_6. Although the sample had been exposed in air for a week, the patterned area was still an effective etch mask. Because the patterned area was very thin and narrow compared to the usual RIE target, the radio-frequency (RF) power of the plasma was lowered from the typical value of 200 W to 50 W in order to prevent the pattern from being destroyed. As indicated by the average cross-section in Figure 5.5(c), the feature height was increased from about 3 nm to more than 20 nm after the RIE process. There was no observable pattern damage, and the pattern linewidth was slightly increased to 80 nm. By increasing the RIE etching time, the feature contrast is expected to be further enhanced.



(a) (b)



(c)

Figure 5.5. A series of boxes written under low-vacuum conditions and imaged by AFM under ambient atmospheric conditions before and after a 8-second RIE etching. The patterns have linewidths of about 80 nm and were around 3 nm in height before the RIE etching took place. After etching, the surface roughness was greatly increased, but the relative feature height was also increased, to about 20 nm, with the patterns intact. The patterning process was performed by George Witzgald. The AFM measurements were performed by Joseph Fu.

Chapter 6

Determination of the Uncertainty Budget for an Interferometer System of Novel Design

6.1 Introduction

The concept of making measurements that are based on atom counting has as one of its fundamental assumptions the notion that at room temperature the lattice constant of the surface lattice of a crystal is the same as the lattice constant of that crystal in bulk. This assumption, however, remains to be experimentally confirmed by traceable high-resolution measurements. To answer this question, a novel design of a high-resolution interferometer is currently being implemented at NIST. The design of this new interferometer aims to achieve sub-angstrom resolution and has been adapted for integration into a UHV STM. The UHV STM should be capable not only of imaging an atomically ordered surface with atomic resolution but also of measuring the displacement of the tip or sample in real time. Measurements with an interferometer are based on the

wavelength of the laser being used, and thus translate into a traceable, direct measurement of the surface lattice spacing.

To achieve sub-angstrom resolution with an interferometer is no trivial task. A conventional interferometer uses a single, fixed-frequency laser and measures displacements by counting interference fringes, which limits the resolution to the fringe spacing, which is given by the formula $\lambda/8$ (where λ is the wavelength). This translates into 80 nm for a He–Ne laser [81]. By utilizing a Zeeman laser, a heterodyne interferometer is capable of linearly subdividing an interference fringe and can achieve a resolution of $\lambda/2048$, which is equivalent to 3 angstroms for the He–Ne Zeeman laser [82]. The new design described here uses a tunable diode laser. During operation, the laser wavelength is dynamically tuned to lock the fringes at the most sensitive position, thereby greatly increasing the resolution (to the sub-angstrom regime).

The interferometer outlined above is currently being implemented. A preliminary experiment was carried out on a graphite sample, which demonstrated its capability to make direct measurements of surface-atom spacings [23]. In this chapter, its design concept, considerations in regard to its operation, and its uncertainty budget are discussed.

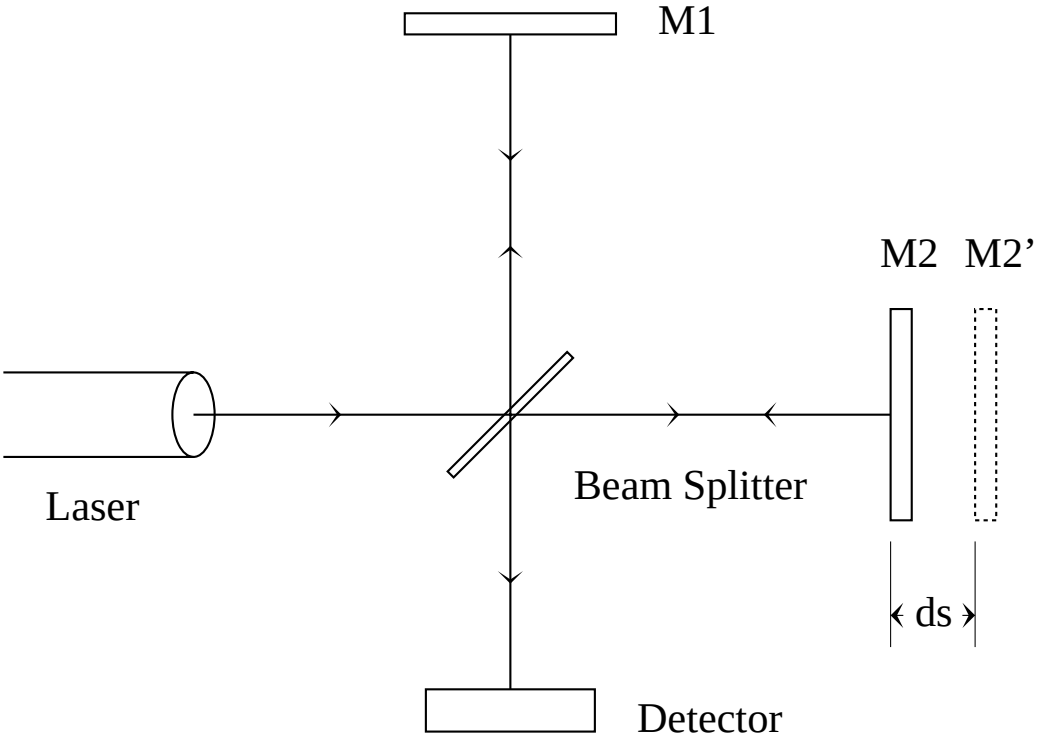


Figure 6.1. The principle of the basic Michelson interferometer.

6.2 Overview of laser interferometer systems

6.2.1 Michelson Interferometer

Most modern displacement-measurement interferometers are based on the configuration of the *Michelson interferometer* [83, 84]. The basic Michelson interferometer is shown in Figure 6.1. A laser beam is directed toward a beam

splitter, which reflects one half of the beam to a fixed mirror (M_1) at 90° and transmits the other half to a movable mirror (M_2). The reflections from the two mirrors are recombined at the beam splitter, and the interference between them is observed by a photodetector. Generally, the two optical paths (the one from the beam splitter to the fixed mirror, and the one from the beam splitter to the movable mirror) are unequal, and the optical-path difference introduces a phase shift between the two reflected beams. The two individual light waves are given by

$$\psi_1 = U_1 e^{i2\pi[(x_1/\lambda)-\nu t]} \tag{6.1}$$

and

$$\psi_2 = U_2 e^{i2\pi[(x_2/\lambda)-\nu t} \tag{6.2}$$

If we define the optical path difference as

$$l = x_2 - x_1 \tag{6.3}$$

and the corresponding phase difference as

$$\phi = 2\pi \frac{l}{\lambda}, \tag{6.4}$$

then the intensity of the resulting (combined) beam is

$$I = U_1^2 + U_2^2 + 2U_1 U_2 \cos\phi \tag{6.5}$$

As the movable mirror is displaced in the direction parallel to the incident beam, the relative phase between the two reflected beams changes and the intensity at the detector increases or decreases.

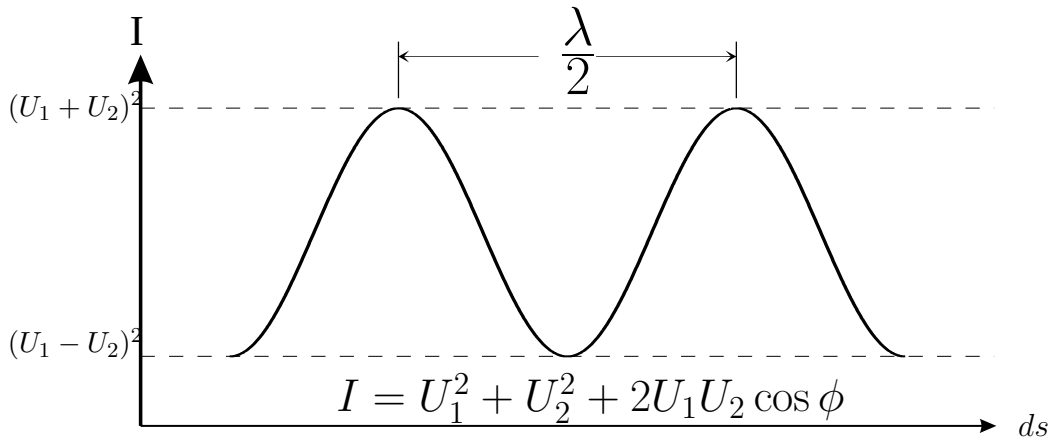


Figure 6.2. The interference intensity of the Michelson interferometer versus the displacement of the moving mirror. Since the laser beam travels twice the length of the measurement arm, the relative phase of the two interfering beams changes by 2π whenever the target mirror displacement changes by $\lambda/2$, in which case the interference intensity is shifted by one cycle (one fringe).

Figure 6.2 shows the calculated light intensity at the detector versus the displacement of mirror M_2. Since the laser beam travels twice the length of the measurement arm, the intrinsic resolution of the Michelson interferometer is $\lambda/2$. Theoretically, it is possible to further subdivide the light intensity at the detector *within* a single cycle, thus increasing the resolution of the measurement. The drawback is that accurate subdivision of a sinusoidal intensity curve is difficult to implement electronically. It is, however, common practice to convert a sinusoidal wave into a square wave and then count the number of quarter fringes [81], which results in a resolution of $\lambda/8$, or 80 nm for a He–Ne laser (of wavelength 633 nm).

6.2.2 Heterodyne laser interferometers

To overcome the limit of the basic Michelson interferometer, modern high-resolution interferometer systems [82] utilize a two-frequency design.

Combining two light waves with different wavelengths (λ_1 and λ_2) and different optical paths (x_1 and x_2), we have

$$\psi_1 = U_1 e^{i2\pi[(x_1/\lambda_1)-\nu_1 t]} \tag{6.6}$$

and

$$\psi_2 = U_2 e^{i2\pi[(x_2/\lambda_2)-\nu_2 t]} \tag{6.7}$$

and the resulting intensity becomes

$$I = U_1^2 + U_2^2 + 2U_1U_2\cos[2\pi((\frac{x_1}{\lambda_1} - \frac{x_2}{\lambda_2}) - (\nu_1 - \nu_2)t)]. \tag{6.8}$$

If the frequency difference $\nu_1 - \nu_2$ is small and constant, the output is an AC signal with fixed frequency, and the optical-path difference is embedded in the phase. The phase of an AC signal can be converted to a linear signal (with the help of a reference signal) and measured electronically, thus greatly increasing the resolution of the measurement. Currently, the best commercial implementation [82] can achieve a resolution of $\lambda/2048$, or 3 angstroms for a laser wavelength of 633 nm.

A common design of a heterodyne laser interferometer is shown in Figure 6.3. The two-frequency laser is commonly obtained by Zeeman splitting of a monochrome laser. The resulting two light waves have a known frequency

difference and mutually perpendicular polarizations. The immediate measurement of the beat signal between the two frequencies (by the photodetector D_1) provides a reference signal. A polarized beam splitter (PBS) is used to force the different polarization components to travel along different optical paths, and the recombination of the two components provides the measurement signal (which is detected by D_2). The phase change introduced by the movement of the retroreflector is measured electronically by comparing the measurement signal to the reference signal. Because of imperfections in the optical coating, different polarizations are inadvertently mixed to some degree, and this limits the resolution of a heterodyne interferometer [85].

6.2.3 The fringe-locking interferometer

Starting with the basic Michelson interferometer, we have derived a new type of interferometer implementation [86, 87]. The optical schematics are the same as in Figure 6.1. Instead of a fixed monochrome laser source, however, a tunable-diode laser source is used. When the movable mirror M_2 undergoes a displacement, the laser frequency is changed in such a way that the number N of wavelengths contained in the optical path difference l remains fixed. This is achieved by using a feedback loop that locks the intensity at the detector at a constant level.

As a result of imposing that condition, we have

$$N = \frac{l}{\lambda} = \frac{l + dl}{\lambda + d\lambda} \tag{6.9}$$

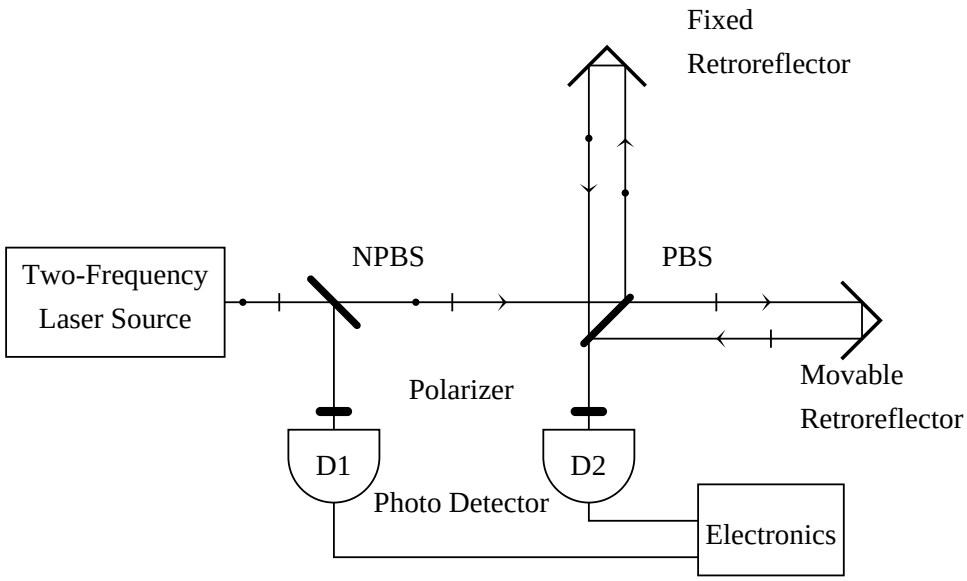


Figure 6.3. The principle of a heterodyne interferometer.

Equivalently,

$$N = \frac{lf}{c} = \frac{(l + dl)(f + df)}{c}, \tag{6.10}$$

where λ is the wavelength of the laser, f is the frequency, and c is the speed of light. Given a change dl in the optical path difference, the laser frequency is shifted by the precise amount df that will keep N fixed. Therefore,

$$\frac{df}{f} = \frac{dl}{l + dl} \approx \frac{dl}{l} \tag{6.11}$$

or

$$dl = \frac{l}{f} df. \tag{6.12}$$

Because the laser beam travels twice the length of the measurement arm, the target mirror displacement $ds = dl/2$. So

$$ds = \frac{l}{2f} df = K df, \tag{6.13}$$

where $K = l/(2f)$ is the measurement coefficient. For a laser with $\lambda = 632.8$ nm and an interferometer path length of $l = 3$ cm, we find that $K = 31$ pm MHz^{-1}. The actual resolution of this design is limited by the performance of the fringe-locking feedback loop and the resolution of the laser-frequency measurement.

6.3 Implementation of the fringe-locking interferometer

6.3.1 Overall system

Figure 6.4 shows the overall design of the fringe-locking interferometer. A tunable laser source—in the current design, a diode laser—is split into two beams. One beam is directed to a Michelson interferometer, and the other beam is used for the frequency measurement. A feedback control setup monitors the intensity at the Michelson interferometer detector and actively tunes the laser so that the intensity at the detector is kept constant.

In this design, the functions of feedback control and frequency measurement are completely separate. To achieve high frequency and high resolution, the feedback utilizes a piezo-electric driver for laser-frequency tuning, similar to the operation of an STM. In STM, the measurement data are simply obtained from the piezo voltage. Because of the non-linear nature of the piezo, the STM measurement is not sufficiently reliable for the metrology. By using a separate frequency-measurement module, the accuracy of the interferometer is independent of the linearity of the piezo. Another benefit of separating the two functions is the ability to troubleshoot each module independently, which significantly eases the implementation.

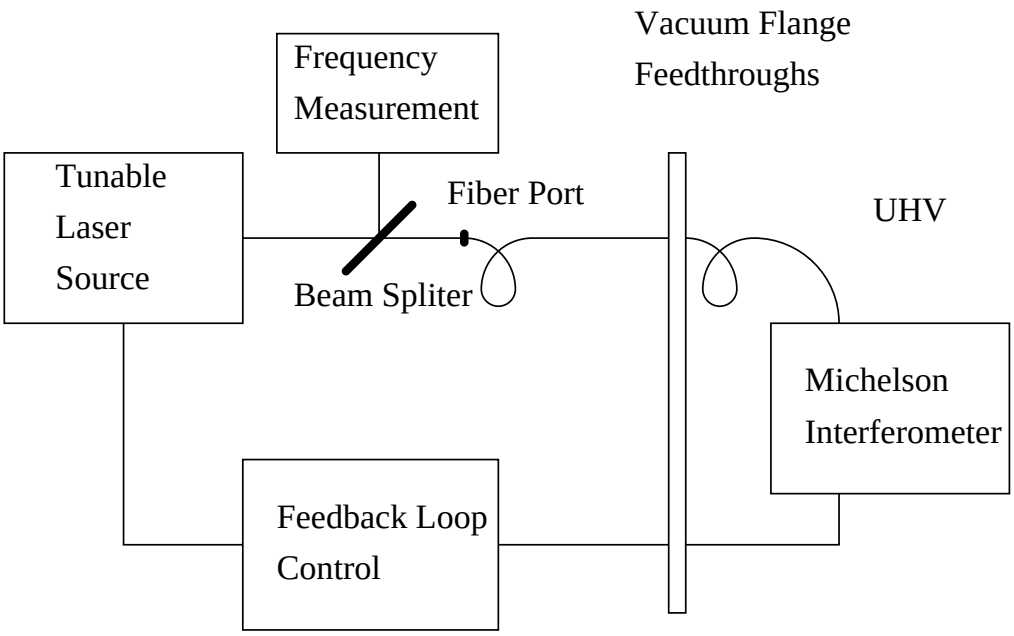


Figure 6.4. The overall design of the fringe-locking interferometer

6.3.2 Laser source selection

To actively lock the number of wavelengths in the optical-path difference, the laser needs to be controlled at, or tuned to, a high frequency. For the purpose of interferometry, it requires a narrow linewidth. In order to accurately determine the laser frequency, a stable reference frequency is required.

To deal with these considerations, an external cavity tunable laser (New Focus [88], Model 6200) is used. The tuning is accomplished by tuning the angle between a pivot mirror and a grating, which is achieved by using a piezoelectric transducer (PZT) stack. The PZT stack can be directly modulated by an external voltage source from -3 V to $+3$ V and can be tuned to a frequency as high as 2 kHz with a 3 dB roll-off point). This gives a tuning range of 60 GHz. In the current implementation, a 10 GHz range is used to optimize the tuning resolution.

Since the tunable diode laser is actively tuned to lock the fringes of the Michelson interferometer, the long-term frequency stability is not important. In the short term (50 ms), its linewidth is less than 300 kHz, which is adequate for 6 pm resolution measurements (assuming the Michelson interferometer is operated at an optical path difference of 2 cm).

The diode laser has a tunable range of 632–637 nm. During operation, it is tuned in the vicinity of the wavelength of a stabilized He–Ne laser, at $\lambda = 632.8$ nm. By beating these two laser beams, what results is a beat signal in the

microwave range (the beat-signal frequency is the difference of the two frequencies), which can be directly measured with a microwave counter.

6.3.3 Laser-frequency measurement

Figure 6.5 illustrates how the frequency is measured. A frequency-stabilized He–Ne laser (Spectra-Physics [89], Model 117A) is used as the reference frequency. The tunable diode laser (TDL) beam is split by a polarized beam splitter. One beam is directed to the Michelson interferometer, and the other beam is mixed with the stabilized He–Ne beam. The mixed beam is filtered with a polarizer and then directed along a single-mode optical fiber. It then feeds into a high-speed photodetector module (New Focus [88], Model 1434). The signal from the photodetector is amplified with a microwave amplifier (Miteq [90], AFS3-00100600-20-ULN), and then it is directly measured by a frequency counter.

There are two main considerations in selecting the frequency counters: range and speed. Because the displacement measurement is linearly related to the laser frequency, the larger the frequency range that the counter can measure, the larger the length range the interferometer can cover. For an optical path difference of 3 cm and a laser wavelength of 632 nm, equation 6.13 gives a 31.6 nm measurement range per GHz of frequency range. For measuring atom spacings, which requires a scanning range of only a few nanometers, a frequency

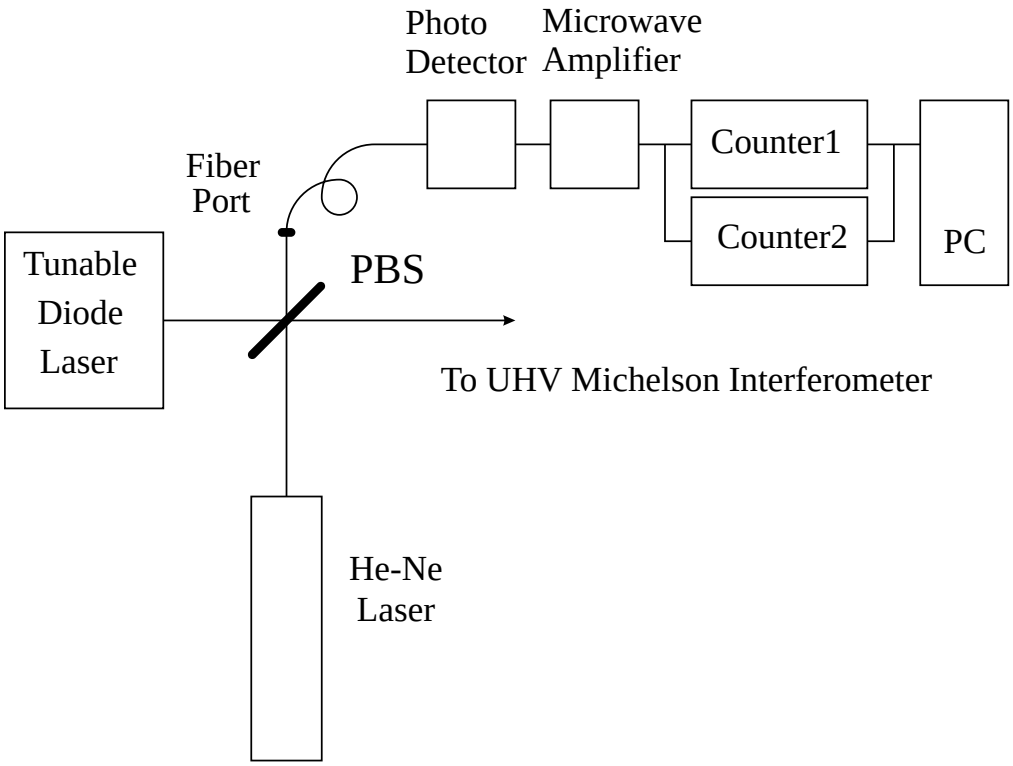


Figure 6.5. The diode laser is beat against a frequency-stabilized He–Ne laser. The beat frequency is measured by two reciprocal counters.

range of a few hundred MHz will suffice. However, it is desirable to make direct measurements of feature dimensions on the order of a few hundred nanometers, which is the typical dimension for current industry applications. Thus, a frequency counter with a range of 10 GHz is preferred.

Another aspect of this interferometer system is that the laser frequency is constantly tuned in order to lock the fringes at the detector, so the laser frequency has to be measured quickly (otherwise, the measurement would be invalid). For low-frequency measurements, the signal can be directly counted by the counter. This offers the highest speed and the best accuracy, but the counter typically cannot measure frequencies above about 300 MHz. For frequencies in the microwave range, typical microwave counters (e.g., the HP 5351B Microwave Frequency Counter [91]) use heterodyne converters by mixing the microwave signal with the harmonics of an internally generated signal, so the difference of the two frequencies falls in the range of an internal direct counter (typically around 100 MHz). Instruments that use this method are able to measure a frequency range higher than 10 GHz (the HP 5351B has a measurement range of 500 MHz to 26.5 GHz). In this approach, a prescan has to be performed to select which harmonic to use for generating a signal that will produce beats within the range of the internal counter. This method works well for signals whose frequency is stable or changes slowly. In our application, the frequency to be measured is rapidly changing, and this method is too slow or otherwise inadequate, on account of the constant need to rescan the harmonics of the internal oscillator.

An alternative method is to use a prescaler. A prescaler is a circuit that divides the input frequency before it is counted by the normal counter. Using a prescaler does not affect the speed of the counter, as no prescan is needed. However, prescalers are not available for microwave signals over 5 GHz. For high-frequency signals, prescalers are not stable and tend to self-oscillate, and the minimum input signal they will accept is rather high. Thus the issue of providing sufficient signal amplification has to be addressed, or peculiar results will be produced.

In the current implementation, a direct counter with a prescaler option (Fluke PM6681 with option PM9625 [92]) is used. This counter has a range of 150 MHz to 4.5 GHz and requires a 25 mV rms minimum input voltage for frequencies 4.2 GHz to 4.5 GHz. As shown in Figure 6.6, the beat frequency is measured while the TDL is being linearly scanned. Frequencies can be measured by the counter anywhere in the range of −4.5 GHz (which is below the He–Ne reference frequency) to 4.5 GHz (which is above the He–Ne reference frequency), with the exception of the range from −200 MHz to 200 MHz, where another direct counter—without a prescaler—is required. The total measurable range is about 9 GHz.

At the time a measurement is made, logic has to be applied to determine which frequency range is being measured. This can be difficult in the case of random measurements. The current interferometer system is designed to be integrated with a scanning tunneling microscope. Fortunately, the direction of

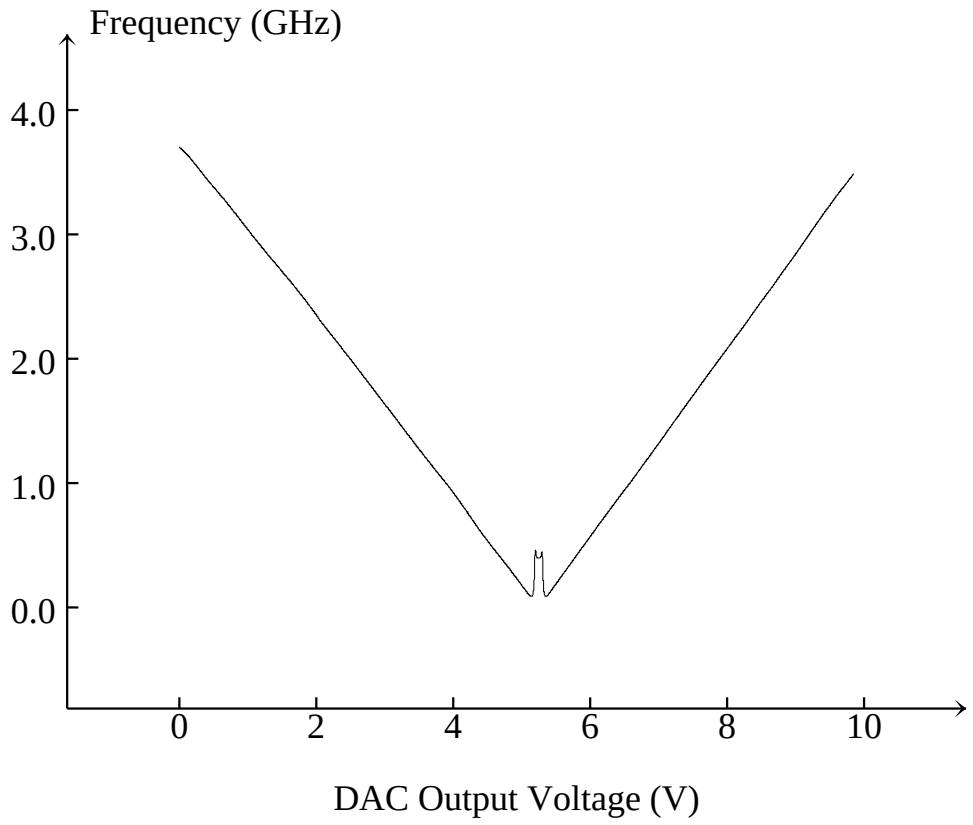


Figure 6.6. The frequency is measured by the counter while the piezo voltage is linearly scanned. There is a 5:1 scaler after the digital to analog converter (DAC), so the actual scanning range is 2 V.

the change in the laser frequency is quite predictable during an STM scan, so the frequency range can usually be determined without ambiguity.

6.3.4 Michelson interferometer unit

The main purpose of designing this interferometer system is to integrate it into a UHV STM. Thus, the design of the Michelson interferometer should be compact, lightweight, rigid, and UHV compatible. In most intensity-based interferometers, fluctuations in the laser-beam intensity are not uncommon. Thus, a proper noise-suppression method also needs to be employed.

The current construction of the interferometer module (which was designed by Lowell Howard) is shown in Figure 6.7. After the TDL beam has been collimated by the optical fiber, it has a polarization of 45° with respect to the vertical position. A calcite crystal is used to split the beam into vertical and horizontal polarizations. The ordinary beam, which has perpendicular polarization, is transmitted through the crystal and then reflected back by a target measurement mirror. This forms the measurement arm. The extraordinary beam, which has horizontal polarization, is reflected back by a silver coating on the end surface of the crystal. This forms the reference arm. The two reflected beams form an elliptically polarized beam. To observe the interference, another calcite crystal—rotated 45° relative to the first one—is used to separate the elliptical beam into two interfering beams, which are detected by separate photodiodes.

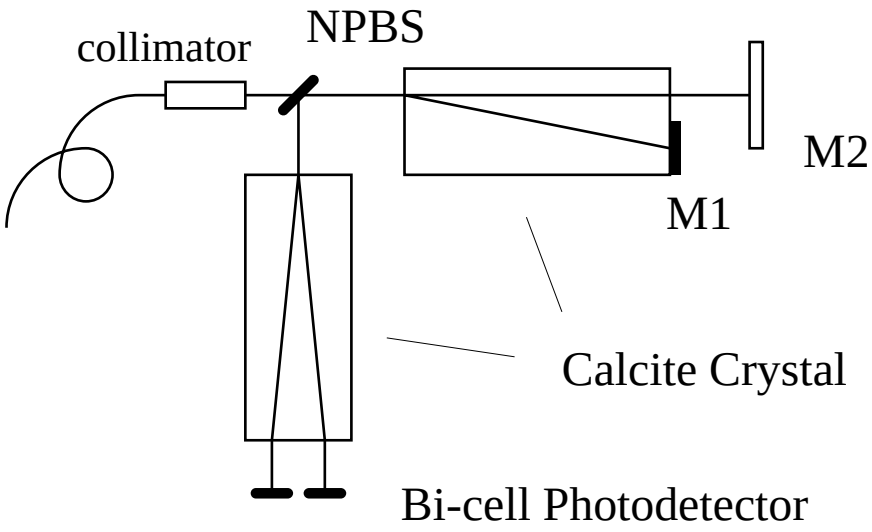


Figure 6.7. The structure of the Michelson interferometer module. The laser beam is split into two beams by a calcite crystal. One beam is reflected off of a reference mirror, which consists of a silver coating on one half of the end surface of the crystal. The other beam is reflected off of a target mirror. A second calcite crystal is used to combine the two polarization components, and the interference signals are detected by two photodiodes.

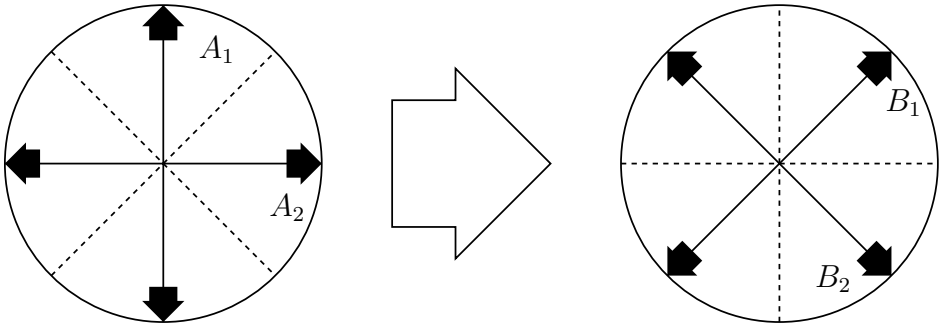


Figure 6.8. The second calcite crystal decomposes the elliptically polarized beam into two linearly polarized beams. If the crystal is aligned at exactly 45°, the two beams will have opposite phases.

The polarization mixing is illustrated in Figure 6.8. Before entering the second calcite crystal, the light waves for the two polarization components are of the form

$$A_1 = U_1 e^{i\phi_1} \tag{6.14}$$

and

$$A_2 = U_2 e^{i\phi_2}, \tag{6.15}$$

where ϕ_1 and ϕ_2 are relative phases. Here the factor which is common to the two waves (namely, $e^{i2\pi(x/\lambda+\nu t)}$) is omitted. After entering the calcite crystal, they are separated into two linearly polarized beams:

$$B_1 = \frac{1}{2}(A_1 + A_2) = \frac{1}{2}(U_1 e^{i\phi_1} + U_2 e^{i\phi_2}) \tag{6.16}$$

and

$$B_2 = \frac{1}{2}(A_1 - A_2) = \frac{1}{2}(U_1 e^{i\phi_1} + U_2 e^{i(\phi_2+\pi)}), \tag{6.17}$$

with intensities of

$$I_1 = U_1^2 + U_2^2 + 2U_1U_2\cos(\Delta\phi) \tag{6.18}$$

and

$$I_2 = U_1^2 + U_2^2 + 2U_1U_2\cos(\Delta\phi + \pi), \tag{6.19}$$

respectively, where $\Delta\phi = \phi_2 - \phi_1$. The difference in the intensities of the two beams is

$$\Delta I = 4U_1U_2\cos(\Delta\phi) \tag{6.20}$$

On comparing this equation with equation 6.8, we see that this intensity difference contains the same information about the phase difference that was introduced by the displacement of the target mirror M_2 in Figure 6.1. However, this configuration doubles the sensitivity. For one thing, the random noise, which is common to I_1 and I_2, is effectively suppressed by taking the difference. Furthermore, by locking the differential signal at the null point (the point where $\cos(\Delta\phi) = 0$), the susceptibility of the measurements to fluctuations in the laser power is eliminated.

6.3.5 Feedback control loop

The signals from the two photodiodes are converted to voltage signals with individual transimpedence amplifiers and are then fed into a differential

amplifier. Since the differential signal is locked at the null point during operation, the differential amplifier can achieve very high gain, which greatly increases the sensitivity of the system. The differential signal is fed into a personal computer (PC), which executes a PID (proportional, integral, and derivative) algorithm [93]. The digital output from the PC is fed into a digital-to-analog converter (DAC), which was made by Lowell Howard using two Burr–Brow integrated circuits, DAC729KH [94], and then directly connected to the TDL's piezo modulation port.

One disadvantage of achieving high gain in the differential amplifier is that when the signal is not locked at the null point, the output is saturated. Thus it is difficult for the feedback loop to return to the null point. This happens when the operation is being started up and when the locking position is being changed (a process known as fringe hopping, which will be discussed later). It is desirable to split the signal and feed it into two differential amplifiers—one with high gain and the other with lower gain—so that the amplified signal is always in the desired range. The control software [95] was written in modular form, so it can accept a set of optimized parameters when searching for the locking position and when actively locking the null point.

Because the feedback loop is designed as a separate unit, it must receive external instructions in order to change the mode of operation, such as locking to the next fringe null point in order to extend the measurement range (fringe hopping). This is implemented using the TCP/IP network. Another PC, which

handles the data acquisition, is able to send commands to the feedback loop in order to allow fringe hopping to be performed.

6.4 Operational considerations

6.4.1 Determining the measurement coefficient

The measurement of the displacement is given by

$$ds = K\,df, \tag{6.21}$$

where the measurement coefficient K is given by

$$K = \frac{l}{2f} = \frac{l}{\lambda}\frac{2\lambda^2}{c} = N\frac{\lambda^2}{2c} \tag{6.22}$$

and N is the number of wavelengths contained in the optical-path difference l.

The measurement coefficient has to be determined before starting the measurement. The method used in determining this coefficient is to fix the target mirror (thus the optical path difference l is constant) and scan the laser frequency. Figure 6.9 plots the differential interference signal against the laser frequency. From one null point to the next calculated null point, N changes by 1/2, so we have

$$N\frac{c}{f_0} = (N + \frac{1}{2})\frac{c}{f_0 + \Delta f} \tag{6.23}$$

Solving for N yields

$$N = \frac{f_0}{2(\Delta f)}, \tag{6.24}$$

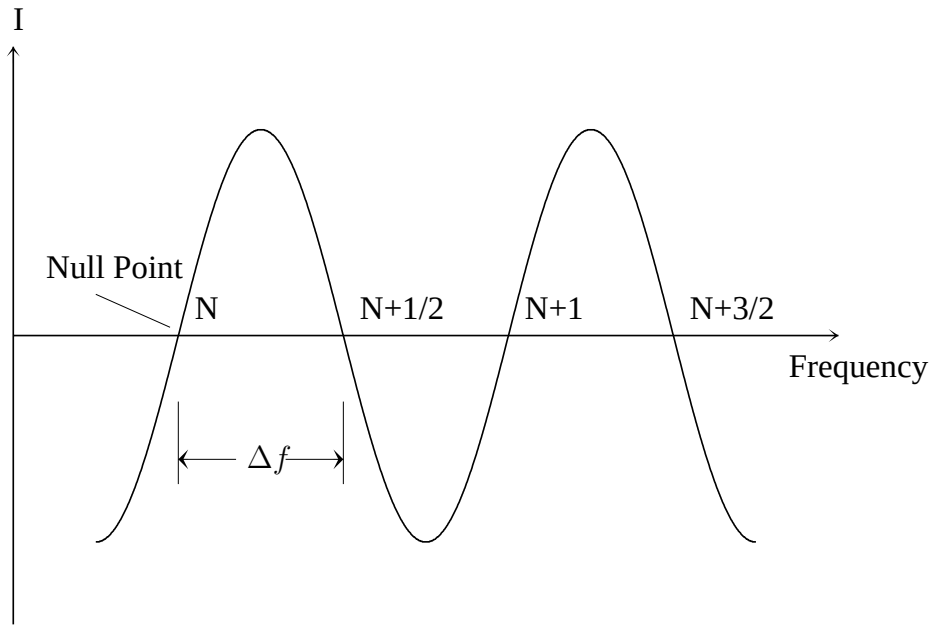


Figure 6.9. The measurement coefficient can be determined by scanning the laser frequency while fixing the optical-path difference. The measurement coefficient is calculated from the wavelength of the laser, together with the difference in frequency between two adjacent null points.

and so

$$K = N\frac{\lambda^2}{2c} = \frac{\lambda}{4(\Delta f)} \tag{6.25}$$

6.4.2 Measurement range without fringe hopping

During the operation, the fringe signal is constantly locked on the null point of one particular fringe, for certain optical-path differences the measurement range is restricted by the tuning range of the tunable diode laser as well as the measurement range of the frequency counter. In the current implementation, the tuning range is 10 GHz. The TDL also has a coarse PZT control through general purpose interface bus (GPIB) that makes it possible to expand the range to 90 GHz. The frequency-measurement range is limited by the range of the frequency counter. Currently, it is 9 GHz. For an optical path difference of 3 cm and a laser wavelength of 632 nm, this gives 31.6 nm per GHz, which translates into a total measurable range of 284 nm.

6.4.3 Fringe hopping

Fringe hopping is the solution to extending the measurement range. By dynamically changing the locking position to the next null point, the measurement constant N is changed by 1/2 and the measurement range is extended by a quarter of a wavelength.

During the fringe hopping, however, the lock on the null point is temporarily lost and the measuring unit has to be properly controlled to avoid reading erroneous data.

If the fringe hopping can be accomplished by use of the usual (fine) PZT

modulation, it can be electronically realized in a few milliseconds. Because of the nature of the laser cavity tuning and the hysteresis property of the PZT, however, after a large jump in voltage the laser frequency will undergo large oscillations, and this will temporarily increase the locking error.

In the case where the coarse PZT control of the TDL has to be used in order to cover the fringe-hopping range, a longer time is spent in waiting for the slow GPIB communication, which translates into even larger gaps during the displacement measurement. If the target mirror is moving at relatively high speed, the system may become unmeasurable, so measurement schemes that would require coarse PZT modulation should be avoided.

6.5 Uncertainty estimation

In this section, we will try to estimate the uncertainty budget of this new interferometer.[1]

Recall the measurement equation:

$$ds = K df \tag{6.26}$$

The direct sources of measurement error are the uncertainty in the laser-frequency measurement and the uncertainty in the measurement

[1]The uncertainty budget has been discussed in [86, 23]. Since the publication of those papers, some changes have been made to the implementation and more testing has been performed. Thus the uncertainty budget evaluated in this work is slightly different from the published values.

coefficient K. Since the basis for equation 6.26 is the assumption that the fringe signal is to be measured at a fixed null point, any deviation from the position of that null point will contribute to the error. Finally, the optical-path difference undergoes two kinds of changes. The first is due to the displacement of the target mirror, which is the intended measurement. The second is due to thermal expansion of the mechanical components, which contributes directly to the measurement error.

The uncertainty estimate is dependent on the length of the optical path difference l. For the following discussion, it is assumed that $l = 3$ cm, which gives $K \approx 21\text{pm} \cdot \text{MHz}^{-1}$ for $\lambda = 632$ nm.

6.5.1 Laser-frequency measurement

The laser frequency is determined by beating the TDL beam with a frequency-stabilized He–Ne laser and measuring its beat frequency using a direct frequency counter. The error in the frequency measurement has three components: instability in the reference laser, instability in the TDL, and uncertainty in the counter measurement.

Since we measure only the changes in the laser frequency, the absolute wavelength of the reference laser is not important, though stability of the frequency during the measurement period is. This particular interferometer was designed mainly for the purpose of measuring STM scans. In STM scanning, a

typical scanning line lasts less than 10 seconds. Over such a short interval, any instability in the He–Ne laser would make a contribution of at most 100 kHz [96] (as quoted from the specification sheet). This translates into an uncertainty of 3.2 pm in the actual measurement.

Since we are constantly tuning the diode laser in order to lock the fringes, the only time at which instability in the TDL contributes to the uncertainty budget is when a frequency measurement is being made. A single frequency is measured in less than 1 ms [97]. According to the specifications of the particular diode laser we are using [98], any instability over a period of 50 ms would contribute at most 300 kHz. As a conservative estimate, the uncertainty in the frequency measurement contributed by this factor is 100 kHz, or 3.2 pm.

The uncertainty introduced by the frequency counter is signal dependent. In the worst case, when the input signal is so weak that the counter is difficult to trigger, the counter reading will be unreliable or erroneous. Whenever the signal is strong enough to trigger the counter, the uncertainty in the counter measurement is

$$\left(\frac{50\text{ps}}{\text{Measuring time}} + (\text{counter timebase error})\right) * (\text{measured frequency}) \text{ [97]}. \quad (6.27)$$

For a measuring time of 1 ms, the counter timebase error is 5×10^{-7} [97]; thus at 4 GHz, its contribution to the uncertainty in the frequency measurement is 2.2 kHz—or, equivalently, 0.07 pm.

6.5.2 Determination of the measurement constant

The measurement coefficient K is given by

$$K = \frac{\lambda}{4(\Delta f)} \tag{6.28}$$

Thus

$$\frac{\delta K}{K} = \sqrt{\frac{\delta\lambda}{\lambda}^2 + \frac{\delta\Delta f}{\Delta f}^2} \tag{6.29}$$

Now $\lambda = 633.4$ nm, $\delta\lambda = 0.02$ nm, $\Delta f = 7.5$ GHz, and $\delta\Delta f$ is less than 400 kHz (as discussed in the previous section). Thus $\delta K/K = 6.2 \times 10^{-5}$. For a displacement of 200 nm, this gives an uncertainty of 12 pm.

In deriving equation 6.13, we made the approximation

$$\frac{dl}{l + dl} \approx \frac{dl}{l}, \tag{6.30}$$

which introduced an error of dl/l. For $dl = 400$ nm, this gives 10^{-5}, which contributes an uncertainty of 2 pm to the frequency measurement.

6.5.3 Fringe null point locking

There is no such thing as a perfect feedback loop; therefore, the error associated with the deviation in the fringe null point has to be considered. This deviation is defined in terms of the equivalent phase difference needed to produce the actual fringe signal (Figure 6.10). For example, if during the measurement the noise in the differential intensity I has an RMS value of δI, then the equivalent phase error is $\delta\phi = \sin^{-1}(\delta I/P)$, where P is the peak intensity. Presumably, $\delta\phi$ is very

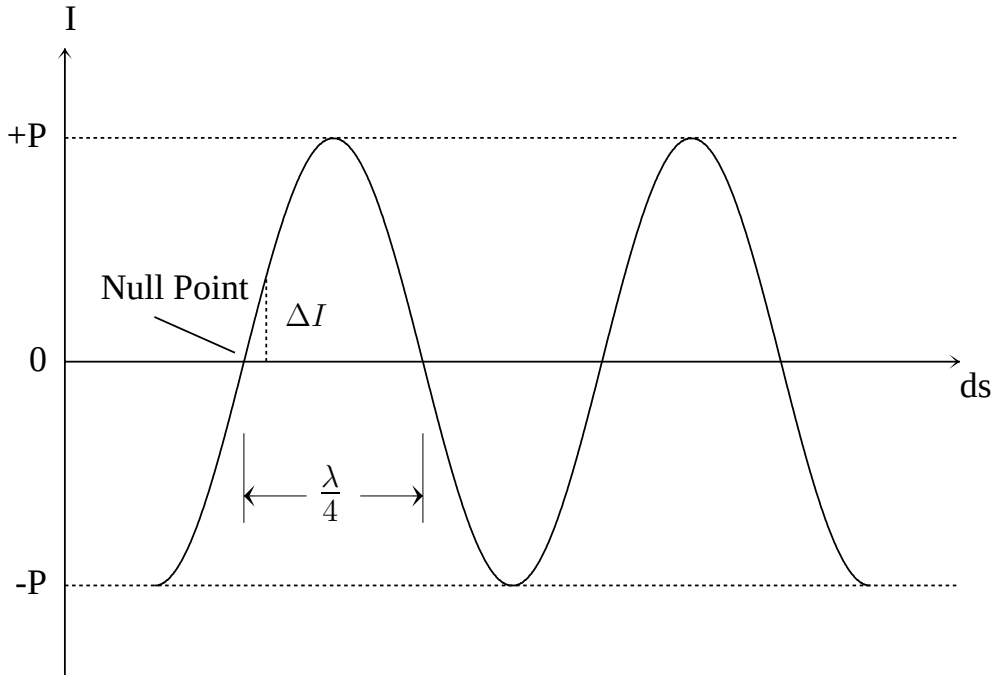


Figure 6.10. For a certain laser frequency, the differential intensity is a sinusoidal function of the target-mirror displacement. The deviation of the actual signal from the null point reflects an error in the displacement measurement.

small, so $\delta\phi = \delta I/P$ and its contribution to the measurement uncertainty is $\delta ds = (\lambda/2)(\delta\phi)/(2\pi)$. For example, suppose that the peak differential intensity is 100 V. (This is a hypothetical value. In a real measurement, any signal above 12 V is clamped by the amplifier. For an estimate of the uncertainty, the hypothetical value should be used.) Assuming that the interferometer is able to lock the fringe within 10 mV (which was a typical value in our tests), the uncertainty is 31 pm.

This error has two possible sources: the interference-fringe detection error and the error caused by the performance of the feedback control. The former causes the feedback loop to lock the fringe at a "false" null point. This "false" null point could be either a real uncertainty or a systematic constant. In the latter case, the system operates at a constant non-null point but the number of wavelengths contained in the optical-path difference is still kept constant, so it doesn't directly contribute to the total error. Because it is not a "true" null point, however, the system is susceptible to laser-power fluctuation.

Both sources are quite complex, and they warrant separate discussion.

6.5.3.1 Fringe-detection error

The first source of this error comes from the noise in the photon detection. This includes the environmental light noise, the intrinsic photodiode noise, and noise added by the subsequent amplification.

The second source comes from the beam alignment. The actual laser beam

used in the interferometer is slightly divergent. In our system, the laser beam typically has a diameter of 2–3 mm. Therefore, the photodiodes detect the integral signal of the interference pattern.

In Figure 6.11(a) the equivalent optical path in vacuum is constructed, and Figure 6.11(b) illustrates the principle of forming interference patterns at the detector. Assuming the light source is a spherical light wave, the interference pattern at the detector is a series of circular rings (Figure 6.12), with each ring representing a different range.

For $L = 13$ cm and $dl = 3$ cm, the radii of the innermost few fringes are $r_1 = 0.94$ mm, $r_2 = 1.32$ mm, $r_3 = 1.62$ mm, and so on. Considering that the diameter of the beam is 2–3 mm (estimated from the size of the laser spot on a paper card), the signal at the detector will always cover a few fringes. Whenever the fringe pattern is localized (that is, the interference pattern is symmetric about the center of the beam), the null point that is measured is the "true" null point, although the more divergent the beam is, the lower the resolution of the fringes will be. If the fringe pattern is not localized (that is, the center of the beam does not overlap the center of the circular fringes), the null point that is measured drifts away from the "true" null point, which introduces a constant systematic error. The interferometer is still able to make accurate measurements. However, because the inter-fringe distance at an off-center position gets very small, the resolution of the fringes will be very poor.

Because of the compact design of the interferometer, it is very difficult to

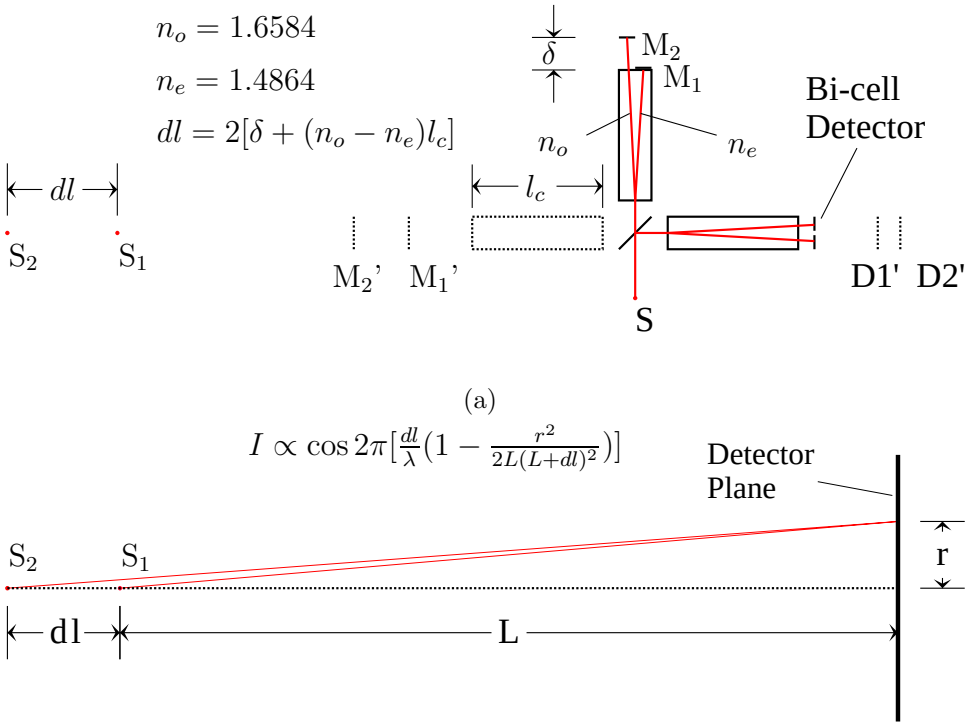
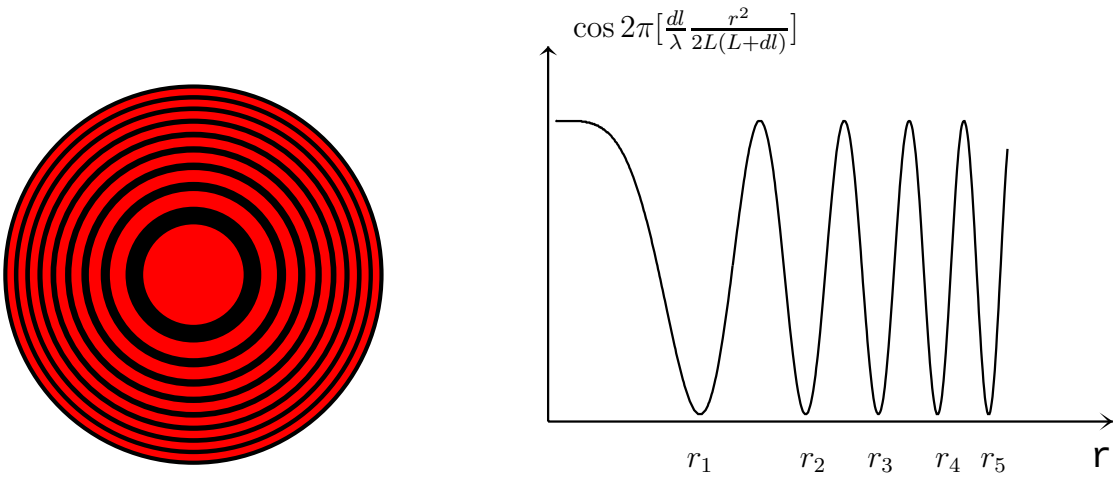


Figure 6.11. (a) Construction of the equivalent optical path in UHV. The optical path difference *dl* consists of two terms: the displacement of mirror M2, and the contribution that arises from the difference in the indices of refraction of the ordinary beam and the extraordinary beam. (b) The simplified equivalent optical path for the Michelson interferometer. The resultant intensity is a function of the displacement of the detector from the center.



(a) (b)

Figure 6.12. The fringe pattern of the interferometer. (a) The localized interference fringe pattern that is produced when a diffuse light source is used. (b) The calculation of the ring radius in the case where dl/λ is an integer.

perfectly align all the optics, and the alignment error is the most significant error. The alignment error can be reduced by placing a approaprate lens after the collimator, which shrinks the size of the laser beam. With further development of robust mirror-adjustment techniques, this error is expected to be minimized.

6.5.3.2 Control-loop error

The performance of the feedback loop is affected by three factors: the loop update speed, the control output resolution, and the efficiency of the algorithm.

The error associated with the loop update speed is dependent on the speed of movement of the measurement target. In a former implementation [86], with the control loop running at 100 Hz, the target mirror underwent a displacement of about 5 nm after each control loop update whenever it was moving at a speed of about 500 nm/s. So even when the control loop was running at its top efficiency and getting the exact value at each update, there was still an average error of about 3 nm. By upgrading to a faster PC, we have been able to run the control loop at 1 kHz with a target speed of less than 100 nm/s, and this gives an error of about 77 pm. As for the measurement of the atom spacings, the target typically moves at a rate of a few nanometers per second, and control-loop error can be further reduced to a few picometers. The control-loop speed is typically limited by the PC data-acquisition speed and the update speed of the laser wavelength following the tuning of the cavity.

Generally, for purposes of feedback control, the accuracy of the output is not

important, as the control adjusts the output in response to the feed-in signal without knowing the actual amplitude of the output. However, the output resolution limits the control-loop efficiency. The output resolution includes the computer digital resolution, the DAC voltage noise, and the piezo responsivity. By using an external, 18-bit digital-to-analog converter, the resolution of the output voltage is less than 10 μV. In our current setup, a resolution of 10 μV corresponds to about 10 kHz in the tuning of the laser frequency, or 0.2 pm in the displacement measurement.

The efficiency of the PID control is signal dependent and is difficult to estimate. Typically, the PID loop tends to work the best during continuous locking, especially when the target mirror is moving in a single direction. Following an event such as a big jump in the target-mirror displacement, or fringe hopping on the part of the controller, a larger error is expected.

6.5.4 Drift in optical-path difference

Because of thermal expansion or mechanical instability, the optical-path difference may change even with the target mirror fixed, and this contributes directly to the total uncertainty of the measurement. In this interferometer system, all the base structures are made of invar (thermal coefficient of expansion 1.26 ppm/K), which minimizes this effect. Moreover, the entire interferometer module is placed in UHV in a controlled temperature environment. Assuming a

Influence parameter	**Estimated value**	**Contribution to the length**
Stability of the He–Ne laser	100 kHz	3.2 pm
Linewidth of the TDL laser	100 kHz	3.2 pm
Frequency-counter uncertainty	2.2 kHz	0.07pm
Measurement coefficient $\frac{\delta K}{K}$	7.2×10^{-5}	14 pm
Fringe-locking uncertainty	Dependent on the measurement range and feedback-loop performance	
Thermal drift in mechanical components	0.1 K/hour	2 pm

Table 6.1. Uncertainty budget of the fringe-locking interferometer

temperature drift of 0.1 K/h, this translates into 3.8 nm/h—or 1 pm/sec—for an optical path difference of $l = 3$ cm. Assuming that a single line scan takes about 2 seconds, it contributes 2 pm to the total uncertainty.

6.6 Summary

Table 6.1 summarizes the uncertainty budget of the fringe-locking interferometer.

The most important undetermined uncertainty is that due to fringe locking

efficiency, which heavily depends on the interference-fringe detection and the speed of the target movement. The former is difficult to estimate at this stage, but in principle, the interference-fringe detection can be improved and there is no intrisic limit set by this factor. As for the latter, for a dimension of 100 nm and a typical sample scanning speed of 100 nm/sec, it contributes an uncertainty of at most 77 pm; and for measurements of atomic spacing, the sample scanning speed is usually below 10 nm/sec, which lowers this uncertainty to below 8 pm.

In summary, the novel design of the fringe-locking interferometer has the potential for making dimensional measurements with sub-angstrom resolution. Although it was designed as a complementary tool for the atom-based metrology project, its application is not limited to nanoscale surface measurement. It could possibly be adapted for use in making displacement measurements in other areas where ultra-high resolution is desired but limited by the conventional tools of metrology.

Appendix A

The Kinetic Monte-Carlo Simulation Program

The program that was used to produce the simulation results in Chapter 3 are listed here as reference. The "param" file is used to set the parameters for the etching, including the etch rates, miscut, size of simulation, and save file information. "Main.c" holds the main function, which simply calls "init_0". Most algorithms of the simulation are implemented in "sim.c".

param

```
#maxrows sets the maximum rows or columns of the simulated surface area
maxrows: 4000
#miscut angle in degree
miscut: 1
#miscut orientation in degree
orient: 120
#simulation continues until extent ML been etched
extent: 8

#Site-specific etch rates                                          10
rate_terr: 1E-7
rate_mstep: 5E-4
rate_hdstep: -1
rate_vdstep: 1E-2
rate_tstep: -1
rate_tterr: -1
rate_point: 1E-1
rate_kink: 1
```

#Initial Pit at Row 1000, Col 1000 with size of 50. All in lattice units.
pit: 1000 1000 50

#filename for saving, the file will be saved as Step.000, Step.0001, ...
name: Step
#Take 8 snapshot saving during the simulation every 1 ML in this example
num: 8
#Or save snapshots every certain etch time (overides last option)
every: 2222 *#10 seconds*

---

param(2)

#maxmem sets the maximum memory the program can use
#In this example, 1.8 Gb maximum gives an buffer 13034 X 21187
#In real size: 6570.97nm X 3793.75nm (the enclosing rectangle)
maxmem: 1800
#The smallest miscut angle used in our experiments
miscut: 0.017
#orient=90 means the miscut is toward <11\bar2>
orient: 107

#Etch rates for various site classes
#This set of rates are used for roughening the initial surface
rate_terr: 1E−19
rate_mstep: 4E−5
rate_hdstep: −1
rate_vdstep: 4E−5
rate_tstep: −1
rate_tterr: −1
rate_point: 5E−3
rate_kink: 4E−5

#The dopant
dopant_ratio: 1e−4
rate_dopant: 1e−5
#And the defect
rate_defect: 10
defect_ratio: 1e−007

#The first roughening etches 4 mono-layers of atoms
extent: 4
#Snapshots will be saved in this name
name: StepA
#making 40 snapshots or every 0.1 mono-layer of etching
num: 40
#the saving file will be in a density of 1 pixel= 2 atom spacing
dumpdensity: 2

```
******** #any thing not understood are discarded silently

#Second stage etching with rate_terr increased.
extent: 2
rate_terr: 1E-9
name: StepB
num: 20

********
..... #more stages omited here

#The final stage of etching, with approarate etch rates.
extent: 45
rate_terr: 1E-10
rate_mstep: 5E-4
rate_hdstep: -1
rate_vdstep: 1E-2
rate_tstep: -1
rate_tterr: -1
rate_point: 1E-1
rate_kink: 1
name: StepG
num: 100
```

---

sim.h

```
#ifndef _Silicon_Etch_Simulation_H
#define _Silicon_Etch_Simulation_H
typedef struct tagSite{
    char spec; //first bit represents defect
    unsigned char depth;
} SITE;

#endif
```

---

main.c

```
#include "sim.h"
#include <stdio.h>

int init_0( char *paramfile );
char *paramfile = "param";
char *logfile = "sisim.log";
FILE *LOG = 0;
```

```
void usage( )
{
    printf( "Usage: sisim [-f paramfile] [-l logfile]\n" );
}

int main( int argc, char **argv )
{
    int i;

    i = 1;
    while ( i < argc ) {
        if ( argv[i][0] == '-' ) {
            switch ( argv[i][1] ) {
              case 'f':
                  if ( argv[i][2] == 0 ) {
                      i++;
                      paramfile = argv[i];
                  }
                  else {
                      paramfile = argv[i] + 2;
                  }
                  printf( "Param file: %s\n", paramfile );
                  break;
              case 'l':
                  if ( argv[i][2] == 0 ) {
                      i++;
                      logfile = argv[i];
                  }
                  else {
                      logfile = argv[i] + 2;
                  }
                  printf( "Log file: %s\n", logfile );
                  break;
              default:
                  printf( "Unknown option: %s\n", argv[i] );
            }
        }
        else
            printf( "Unknown parameter: %s\n", argv[i] );
        i++;
    }
    LOG = fopen( logfile, "w" );
    init_0( paramfile );
//    startetching();
    if ( LOG )
        fclose( LOG );
    return 0;
}
```

```
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>
#include "sim.h"
#include "utility.h"

#define MAXSAVE 1000
#define MAXNAME 20

//#define DEBUG
//#define GLOBALCOUNT
//To add a group, first definition here (it has to be inserted before BULK),
//then add enough algorithm to recognize them – "detector"
//place need to add "detector": resetsite(), etch(), init_2()
//Also need add "input": param init_0()
#define TERR 0
#define MSTEP 1
#define KINK 2
#define VDSTEP 3
#define HDSTEP 4
#define HDPNT 5
#define TSTEP 6
#define TTERR 7
#define DTERR 8
#define DOPANT 9
#define BULK 10
#define NUMSPECS 11

#define FLAG_DEFECT 0x80
#define FLAG_DOPANT 0x40

#define PrintSitesStat(F) {\
    fprintf(F,"\t0 Terrace Sites: \t %d+%d(%d)\n",\
            spechead[TERR],specsize[TERR],specrealsize[TERR]);\
    fprintf(F,"\t1 MonoStep Sites: \t %d+%d(%d)\n",\
            spechead[MSTEP],specsize[MSTEP],specrealsize[MSTEP]);\
    fprintf(F,"\t2 Kink Sites:  \t %d+%d(%d)\n",\
            spechead[KINK],specsize[KINK],specrealsize[KINK]);\
    fprintf(F,"\t3 VDiStep Sites: \t %d+%d(%d)\n",\
            spechead[VDSTEP],specsize[VDSTEP],specrealsize[VDSTEP]);\
    fprintf(F,"\t4 HDiStep Sites: \t %d+%d(%d)\n",\
            spechead[HDSTEP],specsize[HDSTEP],specrealsize[HDSTEP]);\
    fprintf(F,"\t5 HDPoint Sites: \t %d+%d(%d)\n",\
            spechead[HDPNT],specsize[HDPNT],specrealsize[HDPNT]);\
    fprintf(F,"\t6 TriStep Sites: \t %d+%d(%d)\n",\
            spechead[TSTEP],specsize[TSTEP],specrealsize[TSTEP]);\
```

```
    fprintf(F,"\t7 TriTerrace Sites:\t %d+%d(%d)\n",\
            spechead[TTERR],specsize[TTERR],specrealsize[TTERR]);\
    fprintf(F,"\t8 Defect Terrace: \t %d+%d(%d)\n",\
            spechead[DTERR],specsize[DTERR],specrealsize[DTERR]);\
    fprintf(F,"\t9 Dopant Terrace: \t %d+%d(%d)\n",\
            spechead[DOPANT],specsize[DOPANT],specrealsize[DOPANT]);\
}

#define PrintSitesAt(x,y,i,j,n) {\
    for(i=0;i<n;i++){\
        for(j=0;j<n;j++){\
            printf("%d ",allsites[(y-n/2+i)*cols+(x-n/2+j)].depth);\
        }\
        printf("\n");\
    }\
}

#define PrintSites {\
    for(i=0;i<rows;i++){\
        for(j=0;j<cols-i;j++)printf(" ");\
        for(j=0;j<=i;j++){\
            printf("%3d  ",allsites[j*cols+cols-i-1+j].depth);\
        }\
        printf("\n");\
    }\
    for(i=1;i<rows;i++){\
        for(j=0;j<i+1;j++)printf(" ");\
        for(j=0;j<cols-i;j++){\
            printf("%3d  ",allsites[(i+j)*cols+j].depth);\
        }\
        printf("\n");\
    }\
}

#define CheckKink(i,j,index,s) {\
    flag=0;\
    if(findspec(i-1,j+1)==VDSTEP)flag++;\
    if(findspec(i+1,j)==VDSTEP)flag++;\
    if(findspec(i,j-1)==VDSTEP)flag++;\
    if(flag>0)\
        setsite(index,s,KINK);\
    else\
        setsite(index,s,MSTEP);\
}

#define CheckPoint(i,j,index,s) {\
    flag=finddepth(i,j);\
    if(finddepth(i+1,j)<flag){\
        if((finddepth(i-1,j+2)<=flag) && (finddepth(i+1,j-2)<=flag))\
            setsite(index,s,HDSTEP);\
        else\
```

```
            setsite(index,s,HDPNT);\
    }\
    if(finddepth(i−1,j+1)<flag){\
        if((finddepth(i+1,j+1)<=flag) && (finddepth(i−1,j−1)<=flag))\
            setsite(index,s,HDSTEP);\
        else\
            setsite(index,s,HDPNT);\
    }\
    if(finddepth(i,j−1)<flag){\
        if((finddepth(i+2,j−1)<=flag) && (finddepth(i−2,j+1)<=flag))\
            setsite(index,s,HDSTEP);\
        else\
            setsite(index,s,HDPNT);\
    }\
}


#define CheckDiKink(i,j) {\
    if(findspec(i+1,j−1)==MSTEP)resetkink(i+1,j−1);\
    if(findspec(i−1,j)==MSTEP)resetkink(i−1,j);\
    if(findspec(i,j+1)==MSTEP)resetkink(i,j+1);\
}
//if site-B is etched call this macro
#define CheckDiPoint(i,j) {\
    if(findspec(i+1,j+1)==HDSTEP && findspec(i+1,j)==BULK)\
        resetpoint(i+1,j+1);\
    if(findspec(i−1,j+2)==HDSTEP && findspec(i−1,j+1)==BULK)\
        resetpoint(i−1,j+2);\
    if(findspec(i−2,j+1)==HDSTEP && findspec(i−1,j+1)==BULK)\
        resetpoint(i−2,j+1);\
    if(findspec(i−1,j−1)==HDSTEP && findspec(i,j−1)==BULK)\
        resetpoint(i−1,j−1);\
    if(findspec(i+1,j−2)==HDSTEP && findspec(i,j−1)==BULK)\
        resetpoint(i+1,j−2);\
    if(findspec(i+2,j−1)==HDSTEP && findspec(i+1,j)==BULK)\
        resetpoint(i+2,j−1);\
}

//Initially sets up the surface
//Used in Init_2
#define SetupUnit(i,j,index,topdepth) {\
    switch((i−j)%3){\
        case 0:\
            allsites[index].depth=(unsigned char)(topdepth);\
            break;\
        case −1:\
        case 2:\
            allsites[index].depth=(unsigned char)(topdepth+1);\
            break;\
        case 1:\
        case −2:\
```

```
            allsites[index].depth=(unsigned char)(topdepth+3);\
            break;\
    }\
}

int init_1( int maxrows, int orient, double tw );
int init_1m( int maxmem, int orient, double tw );
void init_2( void );
void etch( int index );
void startetching( void );                                          160
void setsite( int index, int from, int to );
void resetkink( int r, int c );
void resetpoint( int r, int c );
int findspec( int r, int c );
int finddepth( int r, int c );

typedef struct tagPIT {
    int x;
    int y;
    int size;                                                       170
    struct tagPIT *next;
} PIT;

PIT *pitslist = 0;

extern FILE *LOG;

int debug = 0;
int stop = 0;
int rows;                                                           180
int cols;
int ux, uy;
int fx, fy;
int xs, ys;
double stepa, stepb, stepc;
double stepd;
SITE *allsites;
int *allposs;
int spechead[NUMSPECS];
int specsize[NUMSPECS];                                             190
int specrealsize[NUMSPECS];
int speccapacity;
double rates[NUMSPECS];
double extent;
double etchtime;
double stops[MAXSAVE];
int nstops;
int nextstop;
char dumpname[MAXNAME];
                                                                    200
#ifdef GLOBALCOUNT
```

```
int totaletch;
int lapetch;
#endif
int etchlap;
double intervaltime = -1.0;
double dumpdensity = 1.0;
double randomize = 0.0;

double defectratio = 0.0;                                                    210
double dopantratio = 0.0;

double latticea = 0.5431021;

#define EMERGENCYSTACKSIZE 1000
int emergencystack[EMERGENCYSTACKSIZE];
int emergencystacktail = 0;
int isemergency = 0;

#ifdef DEBUG                                                                 220
int maxemergencystack = 0;
#endif
void printtime( void )
{
    struct tm *curtime;
    time_t bintime;

    time( &bintime );
    curtime = localtime( &bintime );
    fprintf( LOG, "Current time: %s\n", asctime( curtime ) );                230
}

void dump( char *file )
{
    int i, j;
    int x, y;
    int height;
    int width;
    double ypixelsize;
    double xpixelsize;                                                       240
    float *p;
    int tindex;
    int tdepth;
    int maxdepth = 0;
    int index;
    char buf[100];
    FILE *f;
    int tempint;
    int pos;
                                                                             250
    height = ( rows + cols ) * latticea * sqrt( 6 ) / 12 / dumpdensity;
    width = height * sqrt( 3.0 );
```

```
    ypixelsize = ( rows + cols ) / ( double ) height;
    xpixelsize = ( rows + cols ) / ( double ) width;
    if ( LOG )
        fprintf( LOG, "Dumping into %s image %d X %d, pixelsize=%f(%f nm)\n",
                file, width, height, xpixelsize, dumpdensity );

    p = ( float * ) malloc( sizeof( float ) * width * height );
    for ( i = 0; i < rows * cols; i++ ) {                                    260
        if ( maxdepth < allsites[i].depth ) {
            maxdepth = allsites[i].depth;
        }
    }
    for ( i = 0; i < width * height; i++ )
        p[i] = -maxdepth - 5;
    if ( LOG )
        fprintf( LOG, "Maxdepth= %d\n", maxdepth );

    if ( dumpdensity > 0.3 ) {                                               270
        index = 0;
        for ( i = 0; i < rows; i++ )
            for ( j = 0; j < cols; j++ ) {
                x = ( i + j ) / xpixelsize;
                y = ( rows - i + j ) / ypixelsize;
                if ( x >= width )
                    x = width - 1;
                if ( y >= height )
                    y = height - 1;
                tindex = y * width + x;                                      280
                //So deeper sites get darker
                tdepth = -allsites[index].depth;
                //Register the topmost site
                if ( p[tindex] < tdepth ) {
                    p[tindex] = ( float ) tdepth;
                }
                index++;
            }
    }
    else {                                                                   290
        tindex = 0;
        for ( y = 0; y < height; y++ )
            for ( x = 0; x < width; x++ ) {
                j = ( xpixelsize * x + ypixelsize * y - rows ) / 2;
                i = xpixelsize * x - j;
                index = i * cols + j;
                if ( i >= 0 && i < rows && j >= 0 && j < cols ) {
                    tdepth = -allsites[index].depth;
/*                  if((allsites[index].spec&(unsigned char)0x3f)==VDSTEP){
                        tdepth+=10;                                          300
                    }
                    if((allsites[index].spec&(unsigned char)0x3f)==KINK){
                        tdepth+=20;
```

```
                    }
*/
                    if ( ( allsites[index].spec & ( unsigned char ) 0x80 ) ) { //dislocations
                        tdepth += 20;
                    }
                    p[tindex] = ( float ) tdepth;
                }

                tindex++;
            }
        }
        f = fopen( file, "wb" );
        if ( f == 0 ) {
            printf( "Can't open file %s\n", file );
            return;
        }
        sprintf( buf, "MyData\r\n" );
        fwrite( buf, 1, strlen( buf ), f );

        sprintf( buf, "Aspect: %f\r\n", ( float ) cols / ( float ) rows );
        fwrite( buf, 1, strlen( buf ), f );
        sprintf( buf, "DeadValue: %f\r\n", ( float ) ( -maxdepth - 5 ) );
        fwrite( buf, 1, strlen( buf ), f );
        sprintf( buf, "Pixels: %d\r\n", width );
        fwrite( buf, 1, strlen( buf ), f );
        sprintf( buf, "Lines: %d\r\n", height );
        fwrite( buf, 1, strlen( buf ), f );
        sprintf( buf, "Width: %f\r\n",
                 ( rows + cols ) * latticea / sqrt( 2.0 ) / 2 );
        fwrite( buf, 1, strlen( buf ), f );
        sprintf( buf, "Length: %f\r\n",
                 ( rows + cols ) * latticea / sqrt( 6.0 ) / 2 );
        fwrite( buf, 1, strlen( buf ), f );

        pos = ftell( f );
        sprintf( buf, "Offset: %8d\r\nEnd\r\n", pos + 30 );
        fwrite( buf, 1, strlen( buf ), f );
        fwrite( "zzzzzzzzzzzzzzzzzz\
                zzzzzzzzzzzzzzzzzzzzzzzz", 1, pos + 30 - ftell( f ), f );
        tempint = fwrite( p, sizeof( float ), width * height, f );
#ifdef DEBUG
        if ( tempint != width * height ) {
            printf( "%d/%d written, error %d.\n", tempint, width * height,
                    ferror( f ) );
        }
//      printf("file end offset: %d, should be %d\n",ftell(f),pos+30+sizeof(float)*width*height);
#endif
        fclose( f );
        free( p );
}
```

```
void stopdumping( int index )
{
    char name[MAXNAME];

    sprintf( name, "%s.%03d", dumpname, index );
    printf( "Etch stop %d, dumping %s, etch time %g\n", index, name, etchtime );
    if ( LOG )
        fprintf( LOG, "\nEtch stop %d, dumping %s, etch time %g\n", index, name,
                etchtime );
    if ( LOG )
        PrintSitesStat( LOG )
          dump( name );
}
void savedata( char *file )
{
    FILE *f;
    int i;

    f = fopen( file, "w" );
    if ( f == 0 )
        return;
    fwrite( &cols, sizeof( int ), 1, f );
    fwrite( &rows, sizeof( int ), 1, f );
    for ( i = 0; i < rows * cols; i++ ) {
        fwrite( &allsites[i].depth, 1, 1, f );
    }
    fclose( f );
}
int approx( double t )
{
    int tint;

    if ( t < 1.0 )
        return 1;
    tint = ( int ) t;
    if ( ( t - tint ) > 0.5 )
        return tint + 1;
    else
        return tint;
}

void etchpit( int centerx, int centery, int size )
{
    int i, j;
    int index;

    index = cols * centery + centerx;
    if ( allsites[index].depth == ( allsites[index + 1].depth - 2 ) ) {
        centerx--;
    }
    else if ( allsites[index].depth == ( allsites[index + 1].depth + 3 ) ) {
```

```
        centerx++;
    }
    index = cols * centery + centerx;
    etch( index );
    for ( i = 0; i < size; i++ ) {
        for ( j = 0; j < ( 3 * i + 2 ); j++ ) {
            etch( index - 2 - 3 * i + j * ( cols + 1 ) );
        }
        for ( j = 0; j < ( 3 * i + 3 ); j++ ) {
            etch( index - 3 - 3 * i + j * ( cols + 1 ) );
        }

        for ( j = 0; j < ( 3 * i + 2 ); j++ ) {
            etch( index + ( 2 + 3 * i ) * cols + j * ( 1 - 2 * cols ) );
        }
        for ( j = 0; j < ( 3 * i + 3 ); j++ ) {
            etch( index + 3 * ( i + 1 ) * cols + j * ( 1 - 2 * cols ) );
        }

        for ( j = 0; j < ( 3 * i + 2 ); j++ ) {
            etch( index + ( 2 + 3 * i ) * ( 1 - cols ) + j * ( cols - 2 ) );
        }
        for ( j = 0; j < ( 3 * i + 3 ); j++ ) {
            etch( index + 3 * ( i + 1 ) * ( 1 - cols ) + j * ( cols - 2 ) );
        }
    }

}

void defect( void )
{
    int numdefects;
    int layer;
    unsigned long rnd;
    int i;
    int j;

    layer = rows * cols;
    numdefects = layer * 2 / 3 * defectratio;
    for ( i = 0; i < numdefects; i++ ) {
        rnd = _lrand( );
        j = ( ( double ) rnd / LRAND_MAX ) * layer;
        if ( ( allsites[j].spec & ( unsigned char ) 0x3f ) == TERR ) {
            setsite( j, TERR, DTERR );
        }
        if ( ( allsites[j].spec & ( unsigned char ) 0x3f ) == DOPANT ) {
            setsite( j, DOPANT, DTERR );
        }
        allsites[j].spec |= ( unsigned char ) 0x80;
    }
}
```

```
void dopant( void )
{
    int numdefects;
    int layer;
    unsigned long rnd;
    int i;
    int j;

    layer = rows * cols;
    numdefects = layer * 2 / 3 * dopantratio;
    for ( i = 0; i < numdefects; i++ ) {
        rnd = _lrand( );
        j = ( ( double ) rnd / LRAND_MAX ) * layer;
        if ( ( allsites[j].spec & ( unsigned char ) 0x3f ) == TERR ) {
            setsite( j, TERR, DOPANT );
        }
        allsites[j].spec |= ( unsigned char ) 0x40;
    }
}

void newdopant( void )
{
    int cont = 1;
    unsigned long rnd;
    int layer;
    int j;

    layer = rows * cols;
    while ( cont ) {
        rnd = _lrand( );
        j = ( ( double ) rnd / LRAND_MAX ) * layer;
        if ( allsites[j].spec & ( unsigned char ) 0x40 ) {
            continue;
        }
        else if ( ( allsites[j].spec & ( unsigned char ) 0x3f ) == TERR ) {
            setsite( j, TERR, DOPANT );
        }
        //DTERR remains
        allsites[j].spec |= ( unsigned char ) 0x40;
        break;
    }
}
void nodefect( void )
{
    int i;
    unsigned char s;

    for ( i = 0; i < rows * cols; i++ ) {
        allsites[i].spec &= ( unsigned char ) 0x3f;
        s = allsites[i].spec;
```

```
        if ( s == DTERR || s == DOPANT ) {
            setsite( i, s, TERR );
        }
    }
}

int init_0( char *paramfile )
{
    int i;
    FILE *f;
    char buf[1024];
    int maxrow = -1;
    int maxmem = -1;
    int orient = 0;
    int stepdir;
    int numsave = 0;
    double miscut = 0;
    double ext = 1.0;
    double tw;
    double rate_terr = 1E-7;
    double rate_mstep = 5E-4;
    double rate_hdstep = -1;
    double rate_vdstep = 1E-2;
    double rate_tstep = -1;
    double rate_tterr = -1;
    double rate_kink = 1;
    double rate_point = 0.10;
    double rate_defect = 1E-5;
    double rate_dopant = 1E-5;

    /*
       double pre_rate_terr=1E-7;
       double pre_rate_mstep=5E-4;
       double pre_rate_hdstep=-1;
       double pre_rate_vdstep=1E-2;
       double pre_rate_tstep=-1;
       double pre_rate_tterr=-1;
       double pre_rate_kink=1;
       double pre_rate_point=0.10; */
    PIT *pits;
    int pos;

/*    unsigned long t1;
    double t2;
*/
    int stage = 0;
    int notfinished = 1;
    int newrates;


    f = fopen( paramfile, "r" );
```

```
  if ( f == 0 )
      return 0;
newstage:
  stage++;
  nstops = 0;
  strcpy( dumpname, "dump" );
  newrates = 0;
  while ( fgets( buf, 1024, f ) ) {
      pos = 0;
      if ( getstr( buf, &pos, "maxrows:" ) ) {
          maxrow = getint( buf, &pos );
      }
      if ( getstr( buf, &pos, "maxmem:" ) ) {
          maxmem = getint( buf, &pos );
      }
      if ( getstr( buf, &pos, "orient:" ) ) {
          orient = getint( buf, &pos );
      }
      if ( getstr( buf, &pos, "miscut:" ) ) {
          miscut = getfloat( buf, &pos );
      }
      //actual_rates
      if ( getstr( buf, &pos, "rate_terr:" ) ) {
          rate_terr = getfloat( buf, &pos );
          newrates++;
      }
      if ( getstr( buf, &pos, "rate_mstep:" ) ) {
          rate_mstep = getfloat( buf, &pos );
          newrates++;
      }
      if ( getstr( buf, &pos, "rate_hdstep:" ) ) {
          rate_hdstep = getfloat( buf, &pos );
          newrates++;
      }
      if ( getstr( buf, &pos, "rate_vdstep:" ) ) {
          rate_vdstep = getfloat( buf, &pos );
          newrates++;
      }
      if ( getstr( buf, &pos, "rate_tstep:" ) ) {
          rate_tstep = getfloat( buf, &pos );
          newrates++;
      }
      if ( getstr( buf, &pos, "rate_tterr:" ) ) {
          rate_tterr = getfloat( buf, &pos );
          newrates++;
      }
      if ( getstr( buf, &pos, "rate_kink:" ) ) {
          rate_kink = getfloat( buf, &pos );
          newrates++;
      }
      if ( getstr( buf, &pos, "rate_point:" ) ) {
```

```
    rate_point = getfloat( buf, &pos );
    newrates++;
}
if ( getstr( buf, &pos, "rate_defect:" ) ) {
    rate_defect = getfloat( buf, &pos );
    newrates++;
}
if ( getstr( buf, &pos, "rate_dopant:" ) ) {
    rate_dopant = getfloat( buf, &pos );
    newrates++;
}
if ( getstr( buf, &pos, "defect_ratio:" ) ) {
    defectratio = getfloat( buf, &pos );
    newrates++;
}
if ( getstr( buf, &pos, "dopant_ratio:" ) ) {
    dopantratio = getfloat( buf, &pos );
    newrates++;
}
if ( getstr( buf, &pos, "extent:" ) ) {
    ext = getfloat( buf, &pos );
}
if ( getstr( buf, &pos, "name:" ) ) {
    getname( buf, &pos, dumpname );
    newrates++;
}
if ( getstr( buf, &pos, "save:" ) ) {
    nstops = 0;
    stops[nstops] = getfloat( buf, &pos );
    nstops++;
    while ( getcomma( buf, &pos ) ) {
        stops[nstops] = getfloat( buf, &pos );
        nstops++;
        if ( nstops >= MAXSAVE ) {
            printf( "max save limit reached!" );
            break;
        }
    }
}
if ( getstr( buf, &pos, "num:" ) ) {
    numsave = getint( buf, &pos );
    nstops = 0;
    for ( i = 0; i <= numsave; i++ ) {
        stops[nstops] = i * ext / numsave;
        nstops++;
    }
    newrates++;
}
if ( getstr( buf, &pos, "every:" ) ) {
    intervaltime = getfloat( buf, &pos );
    newrates++;
```

```
        }
        if ( getstr( buf, &pos, "pit:" ) ) {
            pits = ( PIT * ) malloc( sizeof( PIT ) );
            pits->x = getint( buf, &pos );
            pits->y = getint( buf, &pos );
            pits->size = getint( buf, &pos );
            pits->next = pitslist;
            pitslist = pits;
        }
        if ( getstr( buf, &pos, "dumpdensity:" ) ) {
            dumpdensity = getfloat( buf, &pos );
            newrates++;
        }
        if ( getstr( buf, &pos, "randomize:" ) ) {
            randomize = getfloat( buf, &pos );
        }
        if ( getstr( buf, &pos, "********" ) ) {
            if ( stage == 1 )
                goto startetch;
            else
                goto contetch;
        }
    }
    fclose( f );
    notfinished = 0;
    if ( newrates == 0 )
        goto endetch;
    if ( stage > 1 )
        goto contetch;
startetch:
    stepdir = orient - 90;
    if ( stepdir < -30 ) {
        while ( stepdir < 0 )
            stepdir += 360;
    }
    if ( orient >= 330 ) {
        while ( stepdir >= 360 )
            stepdir -= 360;
    }
    if ( miscut < 0.01 ) {
        tw = 0.0;
    }
    else
        tw = 1.0 / sqrt( 3.0 ) / tan( miscut / 180.0 * 3.14159265 ); //in unit of latticea


    if ( LOG ) {
        fprintf( LOG,
                 "Misscut: %.2f, Orientation: %d, Average Terrace Width: %f\n",
                 miscut, orient, tw * latticea );
//      fprintf(LOG,"Maximum Rows: %d\n",maxrow);
```

```
        fprintf( LOG, "Maximum Memory: %d Mb\n", maxrow );
        pits = pitslist;
        if ( pits ) {
            while ( pits ) {
                fprintf( LOG, "Initial pit at (%d,%d) at size %d\n", pits->x,
                        pits->y, pits->size );
                pits = pits->next;
            }
        }
    }
    printtime( );
    if ( maxrow > 0 ) {
        if ( !init_1( maxrow, stepdir, tw ) )
            return 0;
    }
    else if ( maxmem > 0 ) {
        if ( !init_1m( maxmem, stepdir, tw ) )
            return 0;
    }
    else {
        printf( "Must specify the maxrow or maxmem.\n" );
        return 0;
    }
    pits = pitslist;
    if ( pits != 0 ) {
        while ( pits ) {
            printf( "Etching pits (%d,%d)-%d . . . . . .\n", pits->x, pits->y,
                    pits->size );
            etchpit( pits->x, pits->y, pits->size );
            pits = pits->next;
        }
    }
    randomize( );
contetch:
    if ( LOG ) {
        fprintf( LOG, "\nStage %d\n", stage );
        fprintf( LOG, "Relative Etching Rate:\n" );
        fprintf( LOG, "\tTerrace 1-hydride:\t%g\n", rate_terr );
        fprintf( LOG, "\tStep  1-hydride:\t%g\n", rate_mstep );
        fprintf( LOG, "\tStep Horz 2-hydride:\t%g\n", rate_hdstep );
        fprintf( LOG, "\tStep Vert 2-hydride:\t%g\n", rate_vdstep );
        fprintf( LOG, "\tHilllock 2-hydride:\t%g\n", rate_point );
        fprintf( LOG, "\tTerrace 3-hydride:\t%g\n", rate_tterr );
        fprintf( LOG, "\tStep  3-hydride:\t%g\n", rate_tstep );
        fprintf( LOG, "\tDefect terrace:\t%g\n", rate_defect );
        fprintf( LOG, "\tDopant terrace:\t%g\n", rate_dopant );
        fprintf( LOG, "\tKink 1-hydride:\t%g\n", rate_kink );
        fprintf( LOG, "Etch to extent %g layers\n", ext );
        if ( numsave > 0 ) {
            printf( "Saving %d snapshots.\n", numsave );
        }
```

```
        if ( nstops > 0 ) {
            fprintf( LOG, "Dumping to file: %s \n\tAt ", dumpname );
            for ( i = 0; i < nstops - 1; i++ ) {
                fprintf( LOG, "%g, ", stops[i] );
            }
            fprintf( LOG, "%g\n", stops[nstops - 1] );
        }
        if ( intervaltime > 0 ) {                                          770
            fprintf( LOG, "Dumping to file every %g\n", intervaltime );
        }
    }
/*
    if(randomize>0.0){
        printf("Pre-etching %f ......\n",randomize);
    rates[DTERR]=rate_defect;
    rates[BULK]=0;
        rates[TERR]=pre_rate_terr;
        rates[MSTEP]=pre_rate_mstep;                                       780
        rates[HDSTEP]=pre_rate_hdstep;
        rates[VDSTEP]=pre_rate_vdstep;
        rates[TSTEP]=pre_rate_tstep;
        rates[TTERR]=pre_rate_tterr;
        rates[KINK]=pre_rate_kink;
        rates[HDPNT]=pre_rate_point;
        extent=randomize;
        fprintf(LOG,"Randomize With Etch Rates:\n");
        fprintf(LOG,"\tTerrace 1-hydride:\t%g\n",pre_rate_terr);
        fprintf(LOG,"\tStep    1-hydride:\t%g\n",pre_rate_mstep);         790
        fprintf(LOG,"\tStep Horz 2-hydride:\t%g\n",pre_rate_hdstep);
        fprintf(LOG,"\tStep Vert 2-hydride:\t%g\n",pre_rate_vdstep);
        fprintf(LOG,"\tHilllock 2-hydride:\t%g\n",pre_rate_point);
        fprintf(LOG,"\tTerrace 3-hydride:\t%g\n",pre_rate_tterr);
        fprintf(LOG,"\tStep    3-hydride:\t%g\n",pre_rate_tstep);
        fprintf(LOG,"\tKink    1-hydride:\t%g\n",pre_rate_kink);
        fflush(LOG);
        t1=nstops;
        t2=intervaltime;
        nstops=0;                                                          800
        intervaltime=-1;
        startetching();
        nstops=t1;
        intervaltime=t2;
    }
*/
    if ( stage > 1 ) {
        nodefect( );
    }
    if ( defectratio > 0.0 ) {                                             810
        fprintf( LOG, "Adding defect sites with ratio %g.\n", defectratio );
        defect( );
    }
```

```
    if ( dopantratio > 0.0 ) {
        fprintf( LOG, "Adding dopant sites with ratio %g.\n", dopantratio );
        dopant( );
    }

    PrintSitesStat( stdout )
                                                                        820
      rates[TERR] = rate_terr;
    rates[MSTEP] = rate_mstep;
    rates[HDSTEP] = rate_hdstep;
    rates[VDSTEP] = rate_vdstep;
    rates[TSTEP] = rate_tstep;
    rates[TTERR] = rate_tterr;
    rates[KINK] = rate_kink;
    rates[HDPNT] = rate_point;
    rates[DTERR] = rate_defect;
    rates[DOPANT] = rate_dopant;                                        830
    rates[BULK] = 0;
    extent = ext;
    printtime( );

    startetching( );

    printtime( );
    if ( notfinished ) {
        goto newstage;
    }                                                                   840
  endetch:
    return 1;
}

int getnextspec( void )
{
    int i;
    double rnd;
    double RND[NUMSPECS];
    double RND2[NUMSPECS];                                              850
    double sum = 0.0;

#ifdef DEBUG
    if ( rows < 10 ) {
        printf( "Selecting next spec.\n" );
    }
#endif
    for ( i = 0; i < NUMSPECS; i++ ) {
        if ( rates[i] == -1 && specrealsize[i] > 0 )
            return i;                                                   860
    }
    rnd = ( double ) _lrand( );
    for ( i = 0; i < NUMSPECS; i++ ) {
        if ( rates[i] == -1 )
```

```
            RND[i] = 0;
        else
            RND[i] = rates[i] * specrealsize[i];
    }
    for ( i = 0; i < NUMSPECS; i++ ) {
        sum += RND[i];                                                  870
    }
    for ( i = 0; i < NUMSPECS; i++ ) {
        RND2[i] = RND[i] * LRAND_MAX / sum;
    }
    for ( i = 0; i < NUMSPECS; i++ ) {
        rnd = rnd - RND2[i];
        if ( rnd < 0 ) {
            etchtime += 1.0 / RND[i];
            return i;
        }                                                               880
    }
    for ( i = NUMSPECS - 1; i >= 0; i-- ) {
        if ( RND2[i] > 0 ) {
            etchtime += 1.0 / RND[i];
            return i;
        }
    }
    return 0;
}
                                                                        890
int siteselect( int i )
{
    unsigned long rnd;

    while ( 1 ) {
        if ( specsize[i] == 1 ) {
            if ( ( allsites[allposs[spechead[i]]].
                    spec & ( unsigned char ) 0x3f ) == i ) {
                return allposs[spechead[i]];
            }                                                           900
            else {
                printf( "Definitely wrong here: group %d (%x)\n", i,
                        allsites[allposs[spechead[i]]].spec );
            }
        }
        rnd = _lrand( );
        if ( rnd == LRAND_MAX )
            rnd = specsize[i] - 1;
        else
            rnd = ( ( double ) rnd / LRAND_MAX ) * specsize[i];     910
        if ( ( allsites[allposs[spechead[i] + rnd]].
                spec & ( unsigned char ) 0x3f ) == i ) {
            return allposs[spechead[i] + rnd];
        }
        else {
```

```
            allposs[spechead[i] + rnd] = allposs[spechead[i] + specsize[i] - 1];
            specsize[i]--;
        }
    }
}

#define ETCHONE {\
    if(intervaltime>0){\
        if(etchtime>=stoptime){\
            stopdumping(stopindex);\
            stopindex++;\
            stoptime+=intervaltime;\
        }\
    }\
    else{\
        if(i==stopint && j==stopfrac){\
            stopdumping(stopindex);\
            stopindex++;\
            stopint=(int)stops[stopindex];\
            stopfrac=(stops[stopindex]-stopint)*layer;\
        }\
    }\
    if(emergencystacktail>0){\
        isemergency=1;\
        emergencystacktail--;\
        etch(emergencystack[emergencystacktail]);\
    }\
    else{\
        a=getnextspec();\
        b=siteselect(a);\
        etch(b);\
    }\
    j++;\
}


void startetching( )
{
    int i, j;
    int remain;
    int layer;
    int a, b;
    int stopfrac;
    int stopint;
    int stopindex;
    double stoptime = 0.0;

    stopindex = 0;
    layer = rows * cols * 2 / 3;
    etchtime = 0.0;
```

```
#ifdef GLOBALCOUNT
    totaletch = 0;
    lapetch = 0;
#endif                                                                  970
    etchlap = ( extent * layer ) / 100;
    printf( "Start etching %d sites. . .\n", ( int ) ( extent * layer ) );
    i = 0;
    if ( nstops > 0 ) {
        stopint = ( int ) stops[0];
        stopfrac = ( stops[0] - stopint ) * layer;
    }
    else {
        stopint = -1;
    }                                                                   980

    while ( i + 1 <= extent ) {
//      printf("Etching Mono-Layer %d (%d sites)\n",i,layer);
        j = 0;
        while ( j < layer )
            ETCHONE i++;
    }
    remain = ( extent - i ) * layer;
    j = 0;
    while ( j < remain )                                                990
        ETCHONE stopdumping( stopindex );

#ifdef DEBUG
    printf( "Max emergency stack size: %d\n", maxemergencystack );
#endif
}

int init_1m( int maxmem, int stepdir, double tw )
{
    int unitmem, numunits, txs, tys, ttry;                              1000
    int maxsites;
    double ax, ay;

    maxsites = maxmem * 1024 * 1024 / ( sizeof( SITE ) + sizeof( int ) );
    if ( tw == 0 ) {
        ux = ( int ) sqrt( maxsites );
        uy = maxsites / ux;
        fx = fy = 0;
    }
    else {                                                              1010
        //Solving the triangle,
        //ax is the angle facing x(j) axis, and ay faces y(i) axis
        //tw is the height
        if ( stepdir <= 30 ) {
            ay = 30 - stepdir;
            ax = 60 - ay;
            fx = -1;
```

```
        fy = 1;
    }
    else if ( stepdir <= 150 ) {
        ax = 150 - stepdir;
        ay = 120 - ax;
        fx = 1;
        fy = 1;
    }
    else if ( stepdir <= 210 ) {
        ay = 210 - stepdir;
        ax = 60 - ay;
        fx = 1;
        fy = -1;
    }
    else {
        ax = 330 - stepdir;
        ay = 120 - ax;
        fx = -1;
        fy = -1;
    }
    ax = 3.14159265 * ( ax / 180.0 );
    ay = 3.14159265 * ( ay / 180.0 );
    //bond length = a/sqrt(6) or 1 unit in i/j = 1/sqrt(6)
    if ( sin( ax ) < 0.01 ) {
        fy = 0;
        ux = tw * sqrt( 6 );
        uy = maxsites / ux;
    }
    else if ( sin( ay ) < 0.01 ) {
        fx = 0;
        uy = tw * sqrt( 6 );
        ux = maxsites / uy;
    }
    else {
        ux = tw / sin( ay ) * sqrt( 6 );
        uy = tw / sin( ax ) * sqrt( 6 );
    }
}
//circular condition
if ( fx == 0 ) {
    ux = ux + 3 - ux % 3;
}
if ( fy == 0 ) {
    uy = uy + 3 - uy % 3;
}
if ( fx > 0 ) {
    ux = ux - ux % 3 + 2;
}
if ( fx < 0 ) {
    ux = ux - ux % 3 + 1;
}
```

```
if ( fy > 0 ) {
    uy = uy - uy % 3 + 1;
}
if ( fy < 0 ) {
    uy = uy - uy % 3 + 2;
}

unitmem = ( sizeof( SITE ) + sizeof( int ) ) * ux * uy;
numunits = maxmem * 1024 * 1024 / unitmem;
if ( numunits < 1 ) {
    printf( "Sorry, the minimum memory requirement is %s\n", unitmem );
    return 0;
}
txs = tys = 1;
ttry = 1;
while ( ttry ) {
    ttry = 0;
    if ( txs * ux <= tys * uy ) {
        if ( numunits - txs * tys >= tys ) {
            txs++;
            ttry = 1;
        }
        else {
            if ( numunits - txs * tys >= txs ) {
                tys++;
                ttry = 1;
            }
        }
    }
    else {
        if ( numunits - txs * tys >= txs ) {
            tys++;
            ttry = 1;
        }
        else {
            if ( numunits - txs * tys >= tys ) {
                txs++;
                ttry = 1;
            }
        }
    }
}
xs = txs;
ys = tys;
cols = xs * ux;
rows = ys * uy;

if ( LOG ) {
    fprintf( LOG, "\nBuffer Size: %d X %d, Real Size: %.2f X %.2f\n", rows,
             cols, ( rows + cols ) * latticea / sqrt( 2 ) / 2.0,
             ( rows + cols ) * latticea / sqrt( 6 ) / 2.0 );
```

```
        fprintf( LOG, "Periods: %d X %d\n", xs, ys );
        fprintf( LOG, "Ux=%d, Uy=%d\n", ux, uy );
        fprintf( LOG, "Directions: fx=%d, fy=%d\n", fx, fy );
        fprintf( LOG, "\n" );
    }
    init_2(  );
    return 1;
}


int init_1( int maxrows, int stepdir, double tw )
{
    double ax, ay;

    if ( tw == 0 ) {
        ux = uy = maxrows;
        fx = fy = 0;
    }
    else {
        //Solving the triangle,
        //ax is the angle facing x(j) axis, and ay faces y(i) axis
        //tw is the height
        if ( stepdir <= 30 ) {
            ay = 30 - stepdir;
            ax = 60 - ay;
            fx = -1;
            fy = 1;
        }
        else if ( stepdir <= 150 ) {
            ax = 150 - stepdir;
            ay = 120 - ax;
            fx = 1;
            fy = 1;
        }
        else if ( stepdir <= 210 ) {
            ay = 210 - stepdir;
            ax = 60 - ay;
            fx = 1;
            fy = -1;
        }
        else {
            ax = 330 - stepdir;
            ay = 120 - ax;
            fx = -1;
            fy = -1;
        }
        ax = 3.14159265 * ( ax / 180.0 );
        ay = 3.14159265 * ( ay / 180.0 );
        //bond length = a/sqrt(6) or 1 unit in i/j = 1/sqrt(6)
        if ( sin( ax ) < 0.01 ) {
            uy = maxrows;
```

```
            fy = 0;
            ux = tw * sqrt( 6 );
        }
        else if ( sin( ay ) < 0.01 ) {
            ux = maxrows;
            fx = 0;
            uy = tw * sqrt( 6 );
        }
        else {
            ux = tw / sin( ay ) * sqrt( 6 );                                    1180
            uy = tw / sin( ax ) * sqrt( 6 );
        }
    }
    //circular condition
    if ( fx == 0 ) {
        ux = ux + 3 - ux % 3;
    }
    if ( fy == 0 ) {
        uy = uy + 3 - uy % 3;
    }                                                                           1190
    if ( fx > 0 ) {
        ux = ux - ux % 3 + 2;
    }
    if ( fx < 0 ) {
        ux = ux - ux % 3 + 1;
    }
    if ( fy > 0 ) {
        uy = uy - uy % 3 + 1;
    }
    if ( fy < 0 ) {                                                             1200
        uy = uy - uy % 3 + 2;
    }

    xs = ( maxrows + 3 ) / ux;
    ys = ( maxrows + 3 ) / uy;
    cols = xs * ux;
    rows = ys * uy;

    if ( LOG ) {
        fprintf( LOG, "\nBuffer Size: %d X %d, Real Size: %.2f X %.2f\n", rows,  1210
                cols, ( rows + cols ) * latticea / sqrt( 2 ) / 2.0,
                ( rows + cols ) * latticea / sqrt( 6 ) / 2.0 );
        fprintf( LOG, "Periods: %d X %d\n", xs, ys );
        fprintf( LOG, "Ux=%d, Uy=%d\n", ux, uy );
        fprintf( LOG, "Directions: fx=%d, fy=%d\n", fx, fy );
        fprintf( LOG, "\n" );
    }
    init_2(  );
    return 1;
}                                                                               1220
```

```
int translate( int *pr, int *pc )
{
    int diff = 0;

    if ( *pc < 0 ) {
        *pc = cols + *pc;
        diff += 2 * fx * xs;
    }
    if ( *pc >= cols ) {                                    1230
        *pc = *pc - cols;
        diff -= 2 * fx * xs;
    }
    if ( *pr < 0 ) {
        *pr = rows + *pr;
        diff += 2 * fy * ys;
    }
    if ( *pr >= rows ) {
        *pr = *pr - rows;
        diff -= 2 * fy * ys;                                1240
    }
    return diff;
}

void resetkink( int i, int j )
{
    int index;

    if ( j < 0 )
        j = cols + j;                                       1250
    if ( j >= cols )
        j = j - cols;
    if ( i < 0 )
        i = rows + i;
    if ( i >= rows )
        i = i - rows;
    index = i * cols + j;
    setsite( index, MSTEP, KINK );
}
                                                            1260
void resetpoint( int i, int j )
{
    int index;

    if ( j < 0 )
        j = cols + j;
    if ( j >= cols )
        j = j - cols;
    if ( i < 0 )
        i = rows + i;                                       1270
    if ( i >= rows )
        i = i - rows;
```

```
    index = i * cols + j;
    setsite( index, HDSTEP, HDPNT );
}

void resetsite( int i, int j )
{
    int index;
    int flag;                                                    1280
    int s;

    if ( j < 0 )
        j = cols + j;
    if ( j >= cols )
        j = j - cols;
    if ( i < 0 )
        i = rows + i;
    if ( i >= rows )
        i = i - rows;                                            1290
    index = i * cols + j;
    s = ( allsites[index].spec & ( unsigned char ) 0x3f );
    switch ( s ) {
      case BULK:
          if ( allsites[index].depth % 2 == 1 ) {
              CheckKink( i, j, index, BULK )
          }
          else {
              printf( "Strange unexpected situation: bulk at top\n" );
          }                                                      1300
          break;
      case TERR:
      case DTERR:
      case DOPANT:
          setsite( index, s, VDSTEP );
          CheckDiKink( i, j )
            break;
      case MSTEP:
      case KINK:
//            setsite(index,allsites[index].spec,HDSTEP);      1310
          CheckPoint( i, j, index, s )
            break;
      case VDSTEP:
          setsite( index, VDSTEP, TSTEP );
          break;
      case HDSTEP:
          setsite( index, HDSTEP, TTERR );
          break;
      case HDPNT:
          setsite( index, HDPNT, TTERR );                        1320
          break;
      case TSTEP:
          if ( isemergency ) {
```

```
//              isemergency=0;
//              etch(index);
//              isemergency=1;
                allsites[index].depth += 5;
                setsite( index, TSTEP, BULK );
            }
            else {                                                          1330
                etch( index );
#ifdef DEBUG
                //misses the count
                printf
                  ( "rare situation: etch of TSTEP as a result of neighbour etching\n" );
#endif
            }
            break;
        }
}                                                                           1340

void emergency( int i, int j )
{
    int index;

    if ( j < 0 )
        j = cols + j;
    if ( j >= cols )
        j = j - cols;
    if ( i < 0 )                                                            1350
        i = rows + i;
    if ( i >= rows )
        i = i - rows;
    index = i * cols + j;
    emergencystack[emergencystacktail++] = index;
#ifdef DEBUG
    if ( rows < 10 ) {
        printf( "Push into emergency stack [%d]: (%d,%d) - %d\n",
                emergencystacktail - 1, i, j, index );
    }                                                                       1360
    if ( maxemergencystack < emergencystacktail ) {
        maxemergencystack = emergencystacktail;
    }
#endif
}


int finddepth( int r, int c )
{
    int diff;                                                               1370

    diff = translate( &r, &c );
    return allsites[r * cols + c].depth + diff;
}
```

```
int findspec( int r, int c )
{
    translate( &r, &c );
    return allsites[r * cols + c].spec & ( unsigned char ) 0x3f;
}

/*To be checked*/
void defrag( void )
{
    int i, j, k;
    int thead;

    j = 0;
    for ( i = TERR; i < BULK; i++ ) {
        thead = spechead[i];
        spechead[i] = j;
        if ( specsize[i] > 0 ) {
            for ( k = thead; k < thead + specsize[i]; k++ ) {
                if ( ( allsites[allposs[k]].spec & ( unsigned char ) 0x3f ) ==
                     i ) {
                    if ( j < k ) {
                        allposs[j] = allposs[k];
                    }
                    j++;
                }
            }
            specsize[i] = j - spechead[i];
            j++;
        }
        else {
            j++;
        }
    }
    spechead[BULK] = j;
//    if(LOG)PrintSitesStat(LOG)
}

void initspec( void )
{
    int i;

    for ( i = 0; i < NUMSPECS; i++ ) {
        spechead[i] = i;
        specsize[i] = 0;
    }
    speccapacity = rows * cols;
}

void setsite( int index, int from, int to )
{
```

```
    int i;
    int idx;

    if ( allsites[index].depth % 2 == 0 ) {
        if ( ( from == MSTEP || from == HDSTEP || from == TTERR ) &&
             ( to == MSTEP || to == HDSTEP || to == TTERR ) ) {
            printf( "Error: index=%d,from %d to %d\n", index, from, to );
            stop = 1;
        }
    }
    else {
        if ( ( from == TERR || from == VDSTEP || from == TSTEP ) &&
             ( to == TERR || to == VDSTEP || to == TSTEP ) ) {
            printf( "Error: index=%d,from %d to %d\n", index, from, to );
            stop = 1;
        }
    }

    allsites[index].spec &= ( unsigned char ) 0xc0; //preserving the defect label
    allsites[index].spec |= ( unsigned char ) to;
    specrealsize[from]--;
    specrealsize[to]++;
    if ( to == BULK )
        return;
    if ( from == to )
        return;

    allposs[spechead[to] + specsize[to]] = index;
    specsize[to]++;

    for ( i = to + 1; i < NUMSPECS; i++ ) {
        if ( specsize[i] + spechead[i] >= speccapacity - 2 ) {
            printf( "Defragging...\n" );
            defrag( );
            return;
        }
        else if ( spechead[i] == spechead[i - 1] + specsize[i - 1] ) {
            if ( specsize[i] > 0 ) {
                idx = allposs[spechead[i]];
                allposs[spechead[i] + specsize[i]] = idx;
            }
            spechead[i]++;
        }
    }
}

void init_2( )
{
    int flag;
    int k, l;
    int i, j;
```

```
int x, y;
int d;
int topdepth;
int t;                                                          1480
int index;

topdepth = 0;
if ( fx < 0 )
    topdepth += 2 * ( 1 - fx * xs );
if ( fy < 0 )
    topdepth += 2 * ( 1 - fy * ys );

allsites = ( SITE * ) malloc( sizeof( SITE ) * rows * cols );
allposs = ( int * ) malloc( sizeof( int ) * rows * cols );      1490
initspec(  );

if ( LOG )
    fprintf( LOG, "Allocated %d bytes\n\n",
            ( sizeof( SITE ) + sizeof( int ) ) * rows * cols );

//verbosely enumerate each situation
index = 0;
if ( fx == 0 && fy == 0 ) {
    for ( i = 0; i < rows; i++ )                                1500
        for ( j = 0; j < cols; j++ ) {
            SetupUnit( i, j, index, topdepth )
              index++;
        }
}
else if ( fx == 0 ) {
    for ( l = 0; l < ys; l++ ) {
        if ( fy < 0 ) {
            topdepth = 2 * l;
        }                                                       1510
        else
            topdepth = 2 * ( ys - l - 1 );
        for ( i = 0; i < uy; i++ )
            for ( j = 0; j < ux; j++ ) {
                SetupUnit( i, j, index, topdepth )
                  index++;
            }
    }
}
else if ( fy == 0 ) {                                           1520
    for ( k = 0; k < xs; k++ ) {
        if ( fx < 0 ) {
            topdepth = 2 * k;
        }
        else
            topdepth = 2 * ( xs - k - 1 );
        for ( i = 0; i < uy; i++ )
```

```
            for ( j = 0; j < ux; j++ ) {
                index = i * cols + k * ux + j;
                SetupUnit( j, i, index, topdepth )
            }
        }
    }
    else if ( fx == -1 && fy == 1 ) {
        for ( i = 0; i < rows; i++ )
            for ( j = 0; j < cols; j++ ) {
                k = j / ux;
                l = i / uy;
                x = j % ux;
                y = i % uy;
                topdepth = ( k + ( ys - 1 - l ) ) * 2;
                if ( y * ux < x * uy ) {
                    SetupUnit( y, x, index, topdepth + 2 )
                }
                else {
                    SetupUnit( y + 2, x, index, topdepth )
                }
                index++;
            }

    }
    else if ( fx == 1 && fy == 1 ) {
        for ( i = 0; i < rows; i++ )
            for ( j = 0; j < cols; j++ ) {
                k = j / ux;
                l = i / uy;
                x = j % ux;
                y = i % uy;
                topdepth = ( ( xs - 1 - k ) + ( ys - 1 - l ) ) * 2;
                if ( y * ux < ( ux - 1 - x ) * uy ) {
                    SetupUnit( y, x, index, topdepth + 2 )
                }
                else {
                    SetupUnit( y + 2, x, index, topdepth )
                }
                index++;
            }
    }
    else if ( fx == 1 && fy == -1 ) {
        for ( i = 0; i < rows; i++ )
            for ( j = 0; j < cols; j++ ) {
                k = j / ux;
                l = i / uy;
                x = j % ux;
                y = i % uy;
                topdepth = ( ( xs - 1 - k ) + l ) * 2;
                if ( ( uy - 1 - y ) * ux < ( ux - 1 - x ) * uy ) {
                    SetupUnit( y, x, index, topdepth + 2 )
```

```
                }
                else {
                    SetupUnit( y + 2, x, index, topdepth )
                }
                index++;
            }
        }
        else if ( fx == -1 && fy == -1 ) {
            for ( i = 0; i < rows; i++ )
                for ( j = 0; j < cols; j++ ) {
                    k = j / ux;
                    l = i / uy;
                    x = j % ux;
                    y = i % uy;
                    topdepth = ( k + l ) * 2;
                    if ( ( uy - 1 - y ) * ux < x * uy ) {
                        SetupUnit( y, x, index, topdepth + 2 )
                    }
                    else {
                        SetupUnit( y + 2, x, index, topdepth )
                    }
                    index++;
                }
        }

//    PrintSitesAt(100,81,i,j,10)
        index = 0;
        for ( i = 0; i < rows; i++ ) {
            for ( j = 0; j < cols; j++ ) {
                d = allsites[index].depth;
                t = 0;
                if ( d % 2 == 0 ) {
                    if ( finddepth( i, j + 1 ) > ( d + 1 ) )
                        t++;
                    if ( finddepth( i - 1, j ) > ( d + 1 ) )
                        t++;
                    if ( finddepth( i + 1, j - 1 ) > ( d + 1 ) )
                        t++;
                    if ( t == 0 )
                        setsite( index, BULK, TERR );
                    else if ( t == 1 )
                        setsite( index, BULK, VDSTEP );
                    else if ( t == 2 )
                        setsite( index, BULK, TSTEP );
                    else {
                        printf
                          ( "Strange, Not expected floating site after init\n" );
                        allsites[index].depth = ( unsigned char ) ( d + 5 );
                        setsite( index, BULK, BULK );
                    }
                }
```

```
            else {
                if ( finddepth( i, j - 1 ) > ( d - 1 ) )
                    t++;
                if ( finddepth( i + 1, j ) > ( d - 1 ) )
                    t++;
                if ( finddepth( i - 1, j + 1 ) > ( d - 1 ) )
                    t++;
                if ( t == 0 )
                    setsite( index, BULK, BULK );
                else if ( t == 1 )
                    setsite( index, BULK, MSTEP );
                else if ( t == 2 )
                    setsite( index, BULK, HDSTEP );
                else {
                    printf( "Strange, Not expected TTERR\n" );
                    setsite( index, BULK, TTERR );
                }
            }
            index++;
        }
    }
    if ( LOG ) {
        fprintf( LOG, "Sites initilized.\n" );
        PrintSitesStat( LOG )
    }
    for ( k = spechead[MSTEP]; k < spechead[MSTEP] + specsize[MSTEP]; k++ ) {
        index = allposs[k];
        i = index / cols;
        j = index % cols;
        CheckKink( i, j, index, MSTEP )
    }
    for ( k = spechead[HDSTEP]; k < spechead[HDSTEP] + specsize[HDSTEP]; k++ ) {
        index = allposs[k];
        i = index / cols;
        j = index % cols;
        CheckPoint( i, j, index, HDSTEP )
    }
    printf( "Surface initialized. \n" );
    if ( LOG ) {
        fprintf( LOG, "Surface initilized and checked.\n" );
        PrintSitesStat( LOG )
    }
}

void exitsim( )
{
    free( allsites );
    free( allposs );
}

void etch( int index )
```

```
{
    int d, t, flag;
    int i, j;
    int s;
    int td;

#ifdef GLOBALCOUNT
    totaletch++;
    lapetch++;
#endif                                                          1690
    j = index % cols;
    i = index / cols;
    d = allsites[index].depth;
    s = allsites[index].spec & ( unsigned char ) 0x3f;
    allsites[index].spec &= ( unsigned char ) 0xc0;
    allsites[index].spec |= BULK;       //to avoid back loop
    if ( allsites[index].spec & ( unsigned char ) 0x40 ) {
        allsites[index].spec &= ( unsigned char ) 0xbf; //take out the dopant flag
        newdopant( );
    }                                                           1700

#ifdef DEBUG
    if ( rows < 10 ) {
        printf( "Etch site (%d,%d) - %d\n", i, j, index );
    }
#endif

    t = 0;
    if ( debug )
        printf( "etching site: (%d,%d)-d=%d,s=%d\n", i, j, d, s );  1710
    //Terrace upplevel
    if ( d % 2 == 0 ) {
        allsites[index].depth += ( unsigned char ) 5;
        td = finddepth( i, j + 1 ) - ( d + 1 );
        if ( td == 0 )
            resetsite( i, j + 1 );
        else if ( td == -5 ) {
            emergency( i, j + 1 );
        }
#ifdef DEBUG                                                    1720
        if ( td < 0 && td != -5 )
            printf( "Strange(%d) ", td );
#endif
        td = finddepth( i - 1, j ) - ( d + 1 );
        if ( td == 0 )
            resetsite( i - 1, j );
        else if ( td == -5 ) {
            emergency( i - 1, j );
        }
#ifdef DEBUG                                                    1730
        if ( td < 0 && td != -5 )
```

```
            printf( "Strange(%d) ", td );
#endif
        td = finddepth( i + 1, j - 1 ) - ( d + 1 );
        if ( td == 0 )
            resetsite( i + 1, j - 1 );
        else if ( td == -5 ) {
            emergency( i + 1, j - 1 );
        }
#ifdef DEBUG                                                        1740
        if ( td < 0 && td != -5 )
            printf( "Strange(%d) ", td );
#endif
//      if(finddepth(i,j+1)==d+1){ resetsite(i,j+1); }
//      if(finddepth(i-1,j)==d+1){ resetsite(i-1,j); }
//      if(finddepth(i+1,j-1)==d+1){ resetsite(i+1,j-1); }
        if ( isemergency ) {
            isemergency = 0;
#ifdef DEBUG
            d += 1;                                                 1750
            if ( finddepth( i, j + 1 ) <= ( d + 1 ) )
                t++;
            if ( finddepth( i - 1, j ) <= ( d + 1 ) )
                t++;
            if ( finddepth( i + 1, j - 1 ) <= ( d + 1 ) )
                t++;
            if ( t != 2 ) {
#ifdef GLOBALCOUNT
                printf
                  ( "emergency MONOSTEP not right! t=%d, totaletch=%d spec=%d\n",   1760
                    t, totaletch, s );
#else
                printf( "emergency MONOSTEP not right! t=%d\n", t );
#endif
            }
#endif
            CheckKink( i, j, index, s )
        }
        else {
            d += 5;                                                 1770
            if ( finddepth( i, j - 1 ) < ( d ) )
                t++;
            if ( finddepth( i + 1, j ) < ( d ) )
                t++;
            if ( finddepth( i - 1, j + 1 ) < ( d ) )
                t++;
            if ( t == 3 )
                setsite( index, s, BULK );
            else if ( t == 2 ) {
                CheckKink( i, j, index, s )                         1780
            }
            else if ( t == 1 ) {
```

```
                CheckPoint( i, j, index, s )
            }
            else {
#ifdef DEBUG
                printf( "nogood(%d) ", s );
#endif
                setsite( index, s, TTERR );
            }
        }
    }
    //Terrace lowerlevel
    else {
        allsites[index].depth += ( unsigned char ) 1;
        td = finddepth( i, j - 1 ) - ( d - 1 );
        if ( td == 0 )
            resetsite( i, j - 1 );
        else if ( td == -1 ) {
            emergency( i, j - 1 );
        }
#ifdef DEBUG
        if ( td < -1 ) {
            printf( "weird(%d) - %d,%d\n", td, i, j );
            exit( 0 );
        }
#endif
        td = finddepth( i + 1, j ) - ( d - 1 );
        if ( td == 0 )
            resetsite( i + 1, j );
        else if ( td == -1 ) {
            emergency( i + 1, j );
        }
#ifdef DEBUG
        if ( td < -1 ) {
            printf( "weird(%d) - %d,%d\n", td, i, j );
            exit( 0 );
        }
#endif
        td = finddepth( i - 1, j + 1 ) - ( d - 1 );
        if ( td == 0 )
            resetsite( i - 1, j + 1 );
        else if ( td == -1 ) {
            emergency( i - 1, j + 1 );
        }
#ifdef DEBUG
        if ( td < -1 ) {
            printf( "weird(%d) - %d,%d\n", td, i, j );
            exit( 0 );
        }
#endif
//      if(finddepth(i,j-1)==(d-1))resetsite(i,j-1);
//      if(finddepth(i+1,j)==(d-1))resetsite(i+1,j);
```

```
//      if(finddepth(i-1,j+1)==(d-1))resetsite(i-1,j+1);
        if ( isemergency ) {
            isemergency = 0;
#ifdef DEBUG
            d += 1;
            if ( finddepth( i, j + 1 ) <= ( d + 1 ) )
                t++;                                                    1840
            if ( finddepth( i - 1, j ) <= ( d + 1 ) )
                t++;
            if ( finddepth( i + 1, j - 1 ) <= ( d + 1 ) )
                t++;
            if ( t != 2 )
                printf( "emergecy VDSTEP not right! t=%d\n", t );
#endif
            setsite( index, s, VDSTEP );
        }
        else {                                                          1850
            d += 1;
            if ( finddepth( i, j + 1 ) <= ( d + 1 ) )
                t++;
            if ( finddepth( i - 1, j ) <= ( d + 1 ) )
                t++;
            if ( finddepth( i + 1, j - 1 ) <= ( d + 1 ) )
                t++;
            if ( t == 3 ) {
                if ( allsites[index].spec & ( unsigned char ) 0x80 ) {
                    setsite( index, s, DTERR );                         1860
                }
                else if ( allsites[index].spec & ( unsigned char ) 0x40 ) {
                    setsite( index, s, DOPANT );
                }
                else
                    setsite( index, s, TERR );
            }
            else if ( t == 2 )
                setsite( index, s, VDSTEP );
            else if ( t == 1 )                                          1870
                setsite( index, s, TSTEP );
            else {              /*should never happen */
#ifdef DEBUG
                printf( "rare(%d) ", s );
#endif
//              if(s==6)debug=1;
                allsites[index].spec |= ( unsigned char ) s;
                etch( index );
            }
        }                                                               1880
        CheckDiPoint( i, j )
    }
#ifdef DEBUG
    if ( rows < 10 && cols < 10 )
```

```
        PrintSites
#endif
        }
```

utility.h

```
#ifndef UTILITY_H
#define UTILITY_H
double getfloat(char* buffer,int* pos);
int getint(char* buffer,int* pos);
int getstr(char* buffer,int* pos,char* str);
int getname(char* buffer,int* pos,char* namebuffer);
int getcomma(char* buffer,int* pos);
#endif
```

utility.c

```
#include <string.h>

char utiltmpstr[1024];
int getint(char* buffer, int* pos)
{
    int sign=1;
    int i=*pos;
    int result=0;
    while(buffer[i]!=0 && (buffer[i]==' '||buffer[i]=='\t'))i++;
    if(buffer[i]==0)return 0;
    if(buffer[i]=='-'){
        sign=-1;
        i++;
    }
    while(buffer[i]>='0' && buffer[i]<='9')result=result*10+(buffer[i++]-'0');
    *pos=i;
    if(sign==1)return result;
    else return -result;
}

double getfloat(char* buffer,int* pos)
{
    int j;
    int sign=1;
    int i=*pos;
    int index=0;
    int indexsign=1;
    double t;
    double result=0;
    while(buffer[i]!=0 && (buffer[i]==' '||buffer[i]=='\t'))i++;
```

```
    if(buffer[i]==0)return 0;
    if(buffer[i]=='-'){
        sign=-1;
        i++;
    }
    while(buffer[i]>='0' && buffer[i]<='9')result=result*10+buffer[i++]-'0';
    if(buffer[i]=='.'){
        i++;
        t=0.1;
        while(buffer[i]>='0' && buffer[i]<='9')
        {
            result=result+(double)(buffer[i++]-'0')*t;
            t=t*0.1;
        }
    }
    if(buffer[i]=='E' ||buffer[i]=='e'){
        i++;
        if(buffer[i]=='-'){
            indexsign=-1;
            i++;
        }
        while(buffer[i]>='0' && buffer[i]<='9')index=index*10+buffer[i++]-'0';
        if(indexsign==-1){
            for(j=0;j<index;j++)result=result/10;
        }
        else{
            for(j=0;j<index;j++)result=result*10;
        }
    }
    *pos=i;
    if(sign==1)return result;
    else return -result;
}

int getstr(char* buffer,int* pos,char* str)
{
    int len;
    int i=*pos;
    len=strlen(str);
    while(buffer[i]!=0){
        while(buffer[i]!=0 && buffer[i]!=str[0])i++;
        if(buffer[i]==0)
            return 0;
        if(strncmp(buffer+i,str,len)==0){
            *pos=i+len;
            return 1;
        }
        else
            i++;
    }
    return 0;
```

```
}

int getname(char* buffer, int* pos, char* namebuffer)
{
    int i=*pos;
    int j=0;
    while(buffer[i]==' '||buffer[i]=='\t')i++;
    while(buffer[i]!=' '&& buffer[i]!='\t'&&buffer[i]!='\n')
        namebuffer[j++]=buffer[i++];                                    90
    namebuffer[j]='\0';
    *pos=i;
    return j;
}

int getcomma(char* buffer,int *pos)
{
    int i=*pos;
    while(buffer[i]==' '||buffer[i]=='\t')i++;
    if(buffer[i]==','){                                                 100
        *pos=i+1;
        return 1;
    }
    return 0;
}


void hexcode(unsigned char c,char * out)
{
    out[0]=(char)(c>>4);                                                110
    out[1]=(char)(c&0x0f);
    if(out[0]>9){
        out[0]=(char)(out[0]+'a'-10);
    }
    else{
        out[0]=(char)(out[0]+'0');
    }
    if(out[1]>9){
        out[1]=(char)(out[1]+'a'-10);
    }                                                                   120
    else{
        out[1]=(char)(out[1]+'0');
    }
}
```

BIBLIOGRAPHY

[1] Gordon E. Moore. Cramming more components onto integrated circuits. *Electronics*, 38, April 1965.

[2] http://www.intel.com/research/silicon/mooreslaw.htm.

[3] http://www.nano.gov/omb_nifty50.htm.

[4] S. Hong and C. A. Mirkin. A nanoplotter with both parallel and serial writing capabilities. *Science*, 288:1808, 2000.

[5] L. J. Lauhon, M. S. Gudiksen, D. Wang, and C. M. Lieber. Epitaxial core-shell and core-multishell nanowire heterostructures. *Nature*, 420:57, 2002.

[6] A. Vilan, A. Shanzer, and D. Cahen. Molecular control over Au/GaAs diodes. *Nature*, 404:166, 2000.

[7] Semiconductor Industry Association(SIA). *The International Technology Roadmap for Semiconductors (ITRS)*, page 145. 1999.

[8] http://www.nist.gov/public_affairs/nist_mission.htm.

[9] R. M. Silver, C. P. Jensen, V. Tsai, J. Fu, J. Villarrubia, and E. C. Teague. Developing a method to determine linewidth based on counting the atom-spacings across a line. *Proceedings of SPIE*, 3332:441, 1998.

[10] E. S. Fu, X. S. Wang, and E. D. Williams. Characterization of structures fabricated by atomic force microscope lithography. *Surf. Sci.*, 438:58, 1999.

[11] H. Sakaue, S. Fujiwara, S. Shingubara, and T. Takahagi. Atomic-scale defect control on hydrogen-terminated silicon surface at wafer scale. *Appl. Phys. Lett.*, 78:309, January 2001.

[12] M. Schwartzkopff, P. Radojkovic, M. Enachescu, E. Hartmann, and F. Koch. Electron-stimulated desorption of hydrogen from the Si(111) surface by scanning tunneling microscopy. *J. Vac. Sci. Technol. B*, 14:1336, March 1996.

[13] R. S. Becker, G. S. Higashi, Y. J. Chabal, and A. J. Becker. Atomic scale conversion of clean Si(111)-H-1×1 to Si(111)-2×1 by electron stimulated desorption. *Phys. Rev. Lett.*, 65:1917, October 1990.

[14] G. Binnig, H. Rohrer, Ch. Gerber, and E. Weibei. 7×7 reconstruction on Si(111) resolved in real space. *Phys. Rev. Lett.*, 50:120, January 1983.

[15] K. Hata, T. Kimura, S. Ozawa, and H. Shigekawa. How to fabricate a defect free Si(001) surface. *J. Vac. Sci. Technol. A*, 18:1933, 2000.

[16] J. A. Dagata. Device fabrication by scanned probe oxidation. *Science*, 270:1625, December 1995.

[17] E. S. Snow and P. M. Campbell. AFM fabrication of sub-10-nanometer metal-oxide devices with in situ control of electrical properties. *Science*, 270:1639, 1995.

[18] T. C. Shen, C. Wang, G. C. Abeln, J. R. Tucker, J. W. Lyding, P. Avouris, and R. E. Walkup. Atomic-scale desorption through electronic and vibrational-excitation mechanisms. *Science*, 268:1590, 1995.

[19] C. P. Wade and CED Chidsey. Etch-pit initiation by dissolved oxygen on terraces of H-Si(111). *Appl. Phys. Lett.*, 71:1679, September 1997.

[20] G. J. Pietsch, U. Kohler, and M. Henzler. Anisotropic etching versus atomic interaction of atomic steps - scanning tunneling microscopy observations on HF/NH_4F treated Si surfaces. *J. Appl. Phys.*, page 4797, 1993.

[21] H. E. Hessel, A. Feltz, M. Reiter, U. Memmert, and R. J. Behm. Stepflow mechanism versus pit corrosion - scanning tunneling microscopy observations on wet ething of Si(111) by HF solutions. *Chem. Phys. Lett.*, 186:275, 1991.

[22] M. A. Hines. The picture tells the story: using surface morphology to probe etching reactions. *International Rev. Phys. Chem.*, 20:645, October 2001.

[23] R. Silver, H. Zhou, B. Damazo, S. Gonda, J. Jun, C. Jensen, and L. Howard. Atomic-resolution measurement with a new tunable diode laser-based interferometer. *Optical Engin.*, 43:79, 2004.

[24] J. Flidr, Y. C. Huang, T. A. Newton, and M. A. Hines. Extract site specific reaction rates from steady surface morphologies: Kinetic Monte Carlo simulation of aqueous Si(111) etching. *J. Chem. Phys.*, 108:5542, April 1998.

[25] W. Kern. Purifying Si and SiO_2 surfaces with hydrogen peroxide. *Semicond. Int.*, 94, April 1984.

[26] E. Yablonovitch, D. L. Allara, C. C. Chang, T. Gmitter, and T. B. Bright. Unusually low surface-recombination velocity on silicon and germanium surfaces. *Phys. Rev. Lett.*, 57:249, 1986.

[27] D. Graf, M. Grundner, and R. Schulz. Reaction of water with hydrofluoric acid treated silicon(111) and (100) surfaces. *J. Vac. Sci. Technol. A*, 7:808, 1989.

[28] G. S. Higashi, Y. J. Chabal, G. W. Trucks, and K. Raghavachari. Ideal hydrogen termination of Si(111) surfaces. *Appl. Phys. Lett.*, 56:656, February 1990.

[29] U. Neuwald, H. E. Hessel, A. Feltz, U. Memmert, and R. J. Behm. Initial stages of native oxide growth on hydrogen passivated Si(111) surfaces studied by scanning tunneling microscopy. *Appl. Phys. Lett.*, 60:1307, March 1992.

[30] M. Morita, T. Ohmi, E. Hasegawa, M. Kawakami, and M. Ohwada. Growth of native oxide on a silicon surface. *J. Appl. Phys.*, 68:1272, 1990.

[31] R. J. Baierle and M. J. Caldas. Quantum-chemistry study of semiconductor systems: The initial oxidation of the (111)si-h surface. *Int. J. Mod. Phys. B*, 13:2733, 1999.

[32] V. A. Burrows, Y. J. Chabal, G. S. Higashi, K. Raghavachari, and S. B. Christman. Infrared spectroscopy of Si(111) surfaces after HF treatment - hydrogen termination and surface morphology. *Appl. Phys. Lett.*, 53:998, 1988.

[33] P. O. Hahn. *Mater. Res. Soc. Symp. Proc.*, 54:645, 1986.

[34] G. S. Higashi, R. S. Becker, Y. J. Chabal, and A. J. Becker. Comparison of Si(111) surfaces prepared using aqueous solutions of NH_4F versus HF. *Appl. Phys. Lett.*, 58:16568, April 1991.

[35] P. Jakob and Y. J. Chabal. Chemical etching vicinal Si(111)-dependence of surface structure and the hydrogen termination on the pH of the etching solutions. *J. Chem. Phys.*, 95:2897, August 1991.

[36] J. Flidr, Y. C. Huang, and M. A. Hines. An atomistic mechanism for the productions of two- and three-dimensional etch hilllocks on Si(111) surfaces. *J. Chem. Phys.*, 111:6970, October 1999.

[37] V. Le Thanh, D. Bouchier, and G. Hincelin. Low-temperature formation of Si(001)-2×1 surfaces from wet chemical cleaning in NH_4F solution. *J. Appl. Phys.*, 87:3700, 2000.

[38] J. W. Lyding, T.-C. Shen, J. S. Hubacak, J. R. Tucker, and G. C. Abeln. Nanoscale patterning and oxidation of H-passivated Si(100)-2×1 surfaces with an ultrahigh vacuum microscope. *Appl. Phys. Lett.*, 64:2010, April 1994.

[39] E. D. Palik, O. J. Glenbocki, and I. Heard. Study of bias dependent etching of Si in aqueous KOH. *J. Electrochem. Soc.*, 134:404, 1987.

[40] S. P. Garcia, H. L. Bao, M. Manimaran, and M. A. Hines. Measuring the site-specific reactivity of impurities: the pronounced effect of dissolved oxygen on silicon etching. *J. Phys. Chem. B*, 106:8258, August 2002.

[41] H. Fukidome and M. Matsumura. Effect of dissolved oxygen on etching process of Si(111) in NH_3 solution. *Surf. Sci.*, 463, 2000.

[42] W. Kern and D. A. Puotinen. Cleaning solutions based on hydrogen peroxide for use in silicon semiconductor technology. *RCA Rev.*, 31:187, 1970.

[43] T. Takahagi, I. Nagai, A. Ishitani, H. Kuroda, and Y. Nagasawa. The formation of hydrogen passivated silicon single-crystal surfaces using ultraviolet cleaning and HF etching. *J. Appl. Phys.*, 64:3516, 1988.

[44] R. A. Wind, H. Jones, M. J. Little, and M. A. Hines. Orientation-resolved chemical kinetics: Using microfabrication to unravel the complicated chemistry of KOH/Si etching. *J. Phys. Chem. B*, 106:1557, 2002.

[45] R. B. Heimann. *Crystals: Growth, Properties, and Applications*, chapter 4. Springer-Verlag: Berlin, 1982.

[46] M. B. Ives and J. P. Hirth. *J. Chem. Phys.*, 33:517, 1960.

[47] F. C. Frank. *Growth and Perfection of Crystals*, page 4113. Wiley: New York, 1958.

[48] N. Cabrera and D. A. Vermilyea. *Growth and Perfection of Crystals*, page 393. Wiley: New York, 1958.

[49] J. H. Ye, T. H. Bok, J. S. Pan, SFY Li, and J. Y. Lin. Dynamics of hydrides on hydrogen terminated silicon(111)-1x1 surfaces. *J. Phys. Chem. B*, 103:5820, 1999.

[50] Y. C. Huang, J. Flidr, T. A. Newton, and M. A. Hines. Effects of dynamic step-step repulsion and autocatalysis on the morphology of etched Si(111) surfaces. *Phys. Rev. Lett.*, 80:4462, 1998.

[51] H. Ouyang, X. S. Zhao, T. Li, and D. C. Zhang. Direct measurement of the etching rate on Si(111) and silicon dioxide surfaces in NH_4F aqueous solution via atomic force microscopy. *J. Appl. Phys.*, 93:4315, 2003.

[52] D.-J. Liu, J. D. Weeks, and D. Kandel. Current-induced step bending instability on vicianl surfaces. *Phys. Rev. Lett.*, 81:2743, 1998.

[53] D.-J. Liu and J. D. Weeks. Quantitative theory of current-induced step bunching on Si(111). *Phys. Rev. B*, 57:14891, 1998.

[54] S. P. Garcia, H. Bao, and M. A. Hines. Understanding the pH dependence of silicon etching: the importance of dissolved oxygen in buffered HF etchants. *Surface Science*, 541:252, 2003.

[55] T. A. Newton, Y. C. Huang, L. A. Lepak, and M. A. Hines. The site-specific reactivity of isopropanol in aqueous silicon etching: Controlling morphology with surface chemistry. *J. Chem. Phys.*, 111:9125, 1999.

[56] P. Jakob and Y. J. Chabal. Chemical etching of vicinal Si(111): Dependence of the surface structure and the hydrogen termination on the pH of the etching solutions. *J. Chem. Phys.*, 95:2897, 1991.

[57] D. M. Maher and B. Zhang. *J. Vac. Sci. Technol. B*, 12:347, 1994.

[58] V. A. Ukraintsev, R. McGlothin, M. A. Gribelyuk, and H. Edwards. 16:476, 1998.

[59] V. K. Tewary. *Philosophical Magazine*, 80:1445, 2000.

[60] K. Sattler. *Japn. J. Appl. Phys, part 1*, 42:4825, 2003.

[61] J. A. Dagata, J. Schneir, H. H. Harary, C. J. Evans, M. T. Postek, and J. Bennett. Modification of hydrogen passivated silicon by a scanning tunneling microscope operated in air. *Appl. Phys. Lett.*, 56:2001, May 1990.

[62] L. A. Nagahara, T. Thundat, and S. M. Lindsay. Nanolithography on semiconductor surfaces under an etching solution. *Appl. Phys. Lett.*, 57:270, 1990.

[63] N. Barniol, F. Perez-Murano, and X. Aymerich. Modification of HF-treated silicon (100) surfaces by scanning tunneling microscopy in air under imaging conditions. *Appl. Phys. Lett.*, 61:462, 1992.

[64] E. S. Snow, P. M. Campbell, and P. J. McMarr. Fabrication of silicon nanostructures with a scanning tunneling microscope. *Appl. Phys. Lett.*, 63:749, 1993.

[65] H. C. Day and D. R. Allee. Selective area oxidation of silicon with a scanning force microscope. *Appl. Phys. Lett.*, 62:2691, May 1993.

[66] M. Yasutake, Y. Ejjri, and T. Hattori. Modification of silicon surface using atomic force microscope with conducting probe. *Jpn. J. Appl. Phys.*, 32:L1021, 1993.

[67] E. S. Snow and P. M. Campbell. Fabrication of Si nanostructures with atomic force microscope. *Appl. Phys. Lett.*, 64:1932, April 1994.

[68] S. C. Minne, H. T. Soh, Ph. Flueckiger, and C. F. Quate. Fabrication of 0.1 μm metal oxide semiconductor field-effect transistors with the atomic force microscope. *Appl. Phys. Lett.*, 66:703, 1995.

[69] K. Matsumoto, M. Ishii, K. Segawa, Y. Oka, B. J. Vartanian, and J. S. Harris. Room temperature operation of single electron transistor made by the scanning tunneling microscope nanooxidation process for the TiOx/Ti system. *Appl. Phys. Lett.*, 68:34, January 1996.

[70] P. Avouris, T. Hertel, and R. Martel. Atomic force microscope tip-induced local oxidation of silicon: kinetics mechanism and nanofabrication. *Appl. Phys. Lett.*, 71:285, 1997.

[71] J. A. Dagata, T. Inoue, and H. Yokoyama. Understanding scanned probe oxidation of silicon. *Appl. Phys. Lett.*, 73:271, 1998.

[72] R. Garcia, M. Calleja, and F. Perez-Murano. Local oxidation of silicon surface by dynamic force microscopy: Nanofabrication and water bridge formation. *Appl. Phys. Lett.*, 72:2295, 1998.

[73] M. Niwano, J. Kageyama, K. Kurita, K. Kinashi, I. Takahashi, and N. Miyamoto. Infrared spectroscopy study of initial stages of oxidation of hydrogen terminated si surfaces stored in air. *J. Appl. Phys.*, 76:2157, 1994.

[74] H. Ogawa, K. Ishikawa, C. Inomata, and S. Fujimura. Initial stage of native oxide growth on hydrogen terminated silicon (111) surfaces. *J. Appl. Phys.*, 79:472, January 1996.

[75] E. T. Foley, A. F. Kam, J. W. Lyding, and P. Avouris. Cryogenic UHV-STM study of hydrogen and deuterium desorption from Si(100). *Phys. Rev. Lett.*, 80:1336, February 1998.

[76] D. Rogers and H. Nejoh. Fabrication of a metal nanostructure on the Si(111) surface. *J. Vac. Sci. Technol. B*, 17:1323, 1999.

[77] C. P. Jensen, R. Silver, and J. Villarrubia. Experimental results of blind reconstruction of STM tips. *PTB-Bericht*, F-30:119, 1997.

[78] M. Schluter and Marvin. L. Cohen. Nature of conduction-band surface resonances for Si(111) surfaces with and without chemisorbed overlayers. *Phys. Rev. B*, page 716, 1977.

[79] U. Hofer, L. Li, and T. F. Heinz. Desorption of hydrogen from Si(100)2 $\times$ 1 at low coverages: The influence of π -bonded dimers on the kinetics. *Phys. Rev. B*, 45:9485, 1992.

[80] T. A. Takahagi, H. Sakaue, and S. Shingubara. Adsorbed water on a silicon wafer surface exposed to atomophere. *Japanese J. Appl. Phys. part 1*, 40:6198, 2001.

[81] Walter G. Driscoll, editor. *Handbook of Optics*, chapter 16, pages 16–55. McGraw-Hill, New York, 1978.

[82] ZMI 2000 Displace measurement system, Zygo corporation, Laurel Brook Road, Middlefield, CT 06455.

[83] Max Born and Emil Wolf. *Principles of Optics*, pages 300–302. Pergamon Press, 6th edition, 1993.

[84] A. A. Michelson. The relative motion of the earth and the luminiferous aether. *Amer. J. Sci.*, 22:120, 1881.

[85] W. Augustyn and P. Davis. An analysis of polarization mixing errors in distance measuring interferometers. *J. Vac. Sci. Technol. B*, 8:2032, 1990.

[86] S. Gonda, H. Zhou, J. Fu, and R. M. Silver. A new design and uncertainty budget for a metrology UHV-STM used in direct measurements of atom spacings. *Proc. SPIE*, 4608:125, 2001.

[87] The original design was invented by Lowell Howard and Richard Silver inspired from the Fabry-Perot interferometer. To our best knowledge, no similar implementation has been reported.

[88] New Focus, 5215 Hellyer Ave, San Jose, CA 95138.

[89] Spectra Physics, Inc., 1330 Terra Bella Ave, Mountain View, CA 94039.

[90] Miteq, 100 Davids Drive, Hauppauge, NY 11788.

[91] Hewlett-Packard company, 5301 Stevens Creek Boulevard, Santa Clara, CA 95052.

[92] Fluke Corporation, 6920 Seaway Blvd, Everett, WA 98203.

[93] Karl J. Astrom and Tore Hagglund. *PID Controllers: Theory, Design, and Tuning*. The Instrumentation, Systems, and Automation Society, 2 edition, 1995.

[94] Burr-Brown Corporation, Tucson, AZ 85734.

[95] The software for the feedback control and data acquisition is writen in Labview by Hui Zhou and Bradley Damazo.

[96] Specification sheet for Spectra Physics Model 117A Frequency and Intensity Stabilized HeNe Laser obtained from Newport Corporation, 1791 Deere Avenue, Irvine, CA 92606.

[97] Fluke, 7602 EA Almelo, The Netherlands. *Timer/Counter/Analyzers PM6680B & PM6681 Operators Manual*, 1995.

[98] New Focus, Inc., Santa Clara, CA 95051. *The External Cavity Tunable Diode Laser, Model 6200 User's Manual (Rev D).*