

Skew Measurement by Scanning Electrostatic Force Microscopy

David Tadashi Shimizu

Department of Electrical and Computer Engineering
University of Manitoba
Winnipeg, Manitoba, Canada

A thesis submitted to
the Faculty of Graduate Studies
in partial fulfillment of
the requirements for the degree of

Master of Science

Copyright © 1999 by David Tadashi Shimizu



**National Library
of Canada**

**Acquisitions and
Bibliographic Services**

**395 Wellington Street
Ottawa ON K1A 0N4
Canada**

**Bibliothèque nationale
du Canada**

**Acquisitions et
services bibliographiques**

**395, rue Wellington
Ottawa ON K1A 0N4
Canada**

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-45152-6

Canada

**THE UNIVERSITY OF MANITOBA
FACULTY OF GRADUATE STUDIES

COPYRIGHT PERMISSION PAGE**

Skew Measurement by Scanning Electrostatic Force Microscopy

BY

David Tadashi Shimizu

**A Thesis/Practicum submitted to the Faculty of Graduate Studies of The University
of Manitoba in partial fulfillment of the requirements of the degree**

of

MASTER OF SCIENCE

DAVID TADASHI SHIMIZU©1999

Permission has been granted to the Library of The University of Manitoba to lend or sell copies of this thesis/practicum, to the National Library of Canada to microfilm this thesis and to lend or sell copies of the film, and to Dissertations Abstracts International to publish an abstract of this thesis/practicum.

The author reserves other publication rights, and neither this thesis/practicum nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

Abstract

Observation of signals at internal nodes is an important part of integrated circuit failure analysis. However, decreasing circuit dimensions and operating voltages and increasing complexity and clock speeds are making these observations more difficult. The electrostatic force microscope is an instrument that may have the simplicity, non-invasiveness and high spatial, temporal and electric potential resolution needed to probe internal nodes of the latest integrated circuits. This report describes how the software of an existing digital scanning probe microscope controller was modified for scanning electrostatic force microscopy. Several new features were added, including an algorithm to separate the electrostatic force signal from the circuit topography. The report also introduces a signal measurement technique that uses phase modulation to improve the temporal resolution of rectangular sampling pulses, at the cost of information about the measured signal's shape. This technique was used to measure transition times with 280 ps resolution, and propaga-

tion delays with 10 ps resolution on a microstrip transmission line using 1 ns sampling pulses. With the modified digital scanning probe microscope controller, it was used to characterize the gate delay of an integrated inverter, and image the electrostatic force over several integrated inverters at an instant in time.

Acknowledgments

I am greatly indebted to Dr. Doug Thomson and Dr. Greg Bridges for giving me the opportunity to immerse myself in such an innovative, yet relaxed environment. Their selfless hard work and trust in their students make us work for them, not because of them.

I am grateful to everyone who contributed to this thesis. Special thanks to Avid Lemus, for making the first upgrades to the digital SPM controller software; Jocho Nxumalo, for extensively testing the software; Dileepan Joseph, for creatively modelling the phase modulated signal; and Dave Bonin, for an unusually helpful email exchange regarding statistics. Other members of the scanning probe microscopy group and the department's technical staff also deserve recognition for building hardware and writing software used in this research. Their skills and experience were invaluable.

Finally, thanks to my friends and colleagues on campus, for their camaraderie and commiseration; my friends off campus, for their many offers of mindless entertainment;

and my family, for tolerating my sometimes irrational behaviour and for not saying, “Shouldn’t you be done by now?” too often.

This work was supported by the Natural Sciences and Engineering Research Council (NSERC), the Canadian Microelectronics Corporation (CMC), and Micronet.

Contents

List of Figures	x
List of Tables	xiv
List of Symbols	xv
1 Introduction	1
1.1 Motivation for internal node signal measurement.....	1
1.2 Project goals and scope.....	2
1.3 Organization of this report.....	2
2 Internal node signal measurement techniques	4
2.1 Introduction.....	4
2.2 Contact probing	4
2.3 Electron beam testing and photoemission probing.....	5
2.4 Electro-optic probing	7

2.5	Photoconductive sampling.....	8
2.6	Near-field probing.....	8
3	An overview of scanning probe microscopy	10
3.1	Introduction.....	10
3.2	Scanning tunnelling microscopy.....	11
3.3	Scanning force microscopy.....	12
3.3.1	Contact imaging.....	14
3.3.2	Noncontact imaging.....	14
3.3.3	Intermittent contact imaging.....	15
3.4	Scanning force microscope construction.....	16
3.4.1	Cantilever probe.....	16
3.4.2	Piezoelectric scanner.....	17
3.4.3	Cantilever deflection sensor.....	18
3.4.4	Computers and electronic controls	19
3.5	Other techniques.....	20
4	Signal measurement by electrostatic force microscopy	21
4.1	Introduction.....	21
4.2	Theory.....	22
4.3	Probe signal amplitude modulation	23
4.4	Sinusoidal sampling.....	26
4.5	Pulse sampling.....	27
4.6	Performance.....	28

4.6.1	Spatial resolution	28
4.6.2	Temporal resolution	29
4.6.3	Electric potential resolution	30
4.6.4	Invasiveness	32
5	Changes to the digital SPM controller software	33
5.1	Introduction.....	33
5.2	Scanning electrostatic force microscopy	35
5.3	The digital SPM controller hardware	36
5.4	The digital SPM controller software.....	37
5.5	The modified controller software.....	40
5.6	Results and discussion	51
6	Changes to the user interface	56
6.1	Introduction.....	56
6.2	The existing user interface.....	57
6.3	The revised user interface.....	60
6.4	Results and discussion	65
6.5	Future development	67
7	Phase modulated electrostatic force microscopy	70
7.1	Introduction.....	70
7.2	Theory.....	70
7.3	Rectangular pulse sampling.....	73
7.4	Fixed-point measurements.....	77

7.4.1	Experimental setup	77
7.4.2	Results and discussion	79
7.5	One-dimensional scanned measurements	88
7.5.1	Experimental setup	88
7.5.2	Results and discussion	90
7.6	Two-dimensional scanned measurements.....	94
7.6.1	Experimental setup	94
7.6.2	Results and discussion	95
7.7	Suggestions for future work.....	96
8	Conclusions	100
	References	102
	Appendix A DSP program source code	106
	Appendix B SPM image file format	140

List of Figures

Figure 2.1	Electron beam testing.....	5
Figure 2.2	Logic state mapping in EBT [3].....	6
Figure 2.3	External electro-optic probing.	7
Figure 2.4	Photoconductive sampling.	8
Figure 3.1	The probe and sample of a scanning tunnelling microscope.	11
Figure 3.2	Constant height mode (left) versus constant current mode in STM.	12
Figure 3.3	The probe and sample of a scanning force microscope.	13
Figure 3.4	Schematic diagram of a typical scanning force microscope.	16
Figure 3.5	Diagram of a microfabricated SFM cantilever, with probe pointing up-wards.	17
Figure 3.6	Diagram of a piezoelectric tube scanner.	18
Figure 3.7	The “beam-bounce” cantilever deflection detection scheme.	19
Figure 4.1	An EFM probe above a single integrated circuit interconnect.	22

Figure 4.2	Block diagram of the amplitude modulated EFM potential measurement technique.	26
Figure 4.3	A rectangular pulse train.	27
Figure 5.1	Block diagram of a constant force mode SFM feedback control system. .	34
Figure 5.2	Schematic diagram of the original digital SPM controller hardware.	37
Figure 5.3	Schematic diagram of the new SPM controller hardware.	38
Figure 5.4	Raster scanning pattern.	40
Figure 5.5	Feedback points (F) versus image points (I).	41
Figure 5.6	The lift mode technique.	43
Figure 5.7	A typical force versus distance curve.	45
Figure 5.8	An example of a force versus distance curve.	46
Figure 5.9	Finding the offset between the topographical data and the sample surface.	47
Figure 5.10	Flowchart for the modified compensator routine.	50
Figure 5.11	Simultaneously acquired AFM (left) and SRM images of a MOSFET cross-section [26].	55
Figure 5.12	Concurrently acquired AFM (left) and MFM images of a hard disk platter.	55
Figure 6.1	The digital SPM controller's graphical user interface, in review mode. ...	58
Figure 6.2	The digital SPM controller's graphical user interface, in acquire mode. ...	59
Figure 6.3	The new digital SPM controller graphical user interface, in review mode.	66
Figure 6.4	The new digital SPM controller graphical user interface, in acquire mode.	67
Figure 7.1	Equivalent phase modulated (top, middle) and amplitude modulated sampling signals.	73

Figure 7.2	An example of actual (top) and measured circuit signals using phase modulated rectangular pulse sampling.....	74
Figure 7.3	Block diagram of the microstrip transmission line experimental setup.....	77
Figure 7.4	Photograph of the electrostatic force microscope and optical microscope [22].	80
Figure 7.5	Photograph of the EFM cantilever probe transmission line (triangular ground plane facing camera) and an IC under test [22]. The probe itself is too small to be seen here.	81
Figure 7.6	Transmission line signal measured by oscilloscope.	81
Figure 7.7	Sampling signals with a 50 ps delay measured by oscilloscope.....	82
Figure 7.8	Phase modulated EFM measurements plotted against	83
Figure 7.9	Effective sampling signals for various phase modulations.....	84
Figure 7.10	Expected cantilever vibrations for various sampling signal phase modulations.....	84
Figure 7.11	Transmission line propagation delay measurements, smoothed with an 18 ps moving average.	86
Figure 7.12	Straight line fits to the transitions in EFM propagation delay measurements.	87
Figure 7.13	Propagation delays along a microstrip transmission line.....	87
Figure 7.14	Photograph of the cantilever probe and integrated circuit under test [22].	89
Figure 7.15	Diagram of the inverter chain and test pads.....	89
Figure 7.16	Linear topographic (left) and electrostatic force measurements versus time.	91
Figure 7.17	Average phase modulated EFM measurements for each interconnect.	92
Figure 7.18	Average topographic and electrostatic force signals during the 0.8 ns interval.....	93
Figure 7.19	Topographic (left) and electrostatic force images of the inverter chain. ...	95

Figure 7.20	Phase modulated sawtooth sampling signals (top, middle) and the equivalent amplitude modulated sampling signal.	97
Figure 7.21	Equivalent pulse width modulated (top, middle) and amplitude modulated sampling signals.....	98
Figure 7.22	Pulse width modulation using modified phase modulation equipment.	99

List of Tables

Table 5.1	Compensator routine execution time during forward passes.	52
Table B.1.	Revised INF file format.	141

List of Symbols

ADC	analog-to-digital converter
AFM	atomic force microscope/microscopy
DAC	digital-to-analog converter
DSP	digital signal processor
EBT	electron beam tester/testing
EFM	electrostatic force microscope/microscopy
FWHM	full-width half-maximum
IC	integrated circuit
ISR	interrupt service routine
MFM	magnetic force microscope/microscopy
PC	personal computer
PI	proportional-integral

SFM	scanning force microscope/microscopy
SPM	scanning probe microscope/microscopy
SRM	scanning resistance microscope/microscopy
STM	scanning tunnelling microscope/microscopy

1

Introduction

1.1 Motivation for internal node signal measurement

Testing is an important part of integrated circuit (IC) development. After an IC is designed and fabricated, it is tested to verify its functionality and characterize its performance. Its design and manufacture can then be revised to increase yield and quality.

If an IC fails, the defect must be localized and the failure mechanism determined if improvements are to be made. Measurement of signals at internal nodes of integrated circuits is clearly a useful ability here. The high complexity of modern ICs makes observation of internal nodes through external pins difficult. Therefore, an instrument capable of making direct measurements of internal node signals would be a benefit to IC failure analysis.

In today's environment of shrinking devices, increasing clock speeds, and decreasing operating voltages, such an instrument must have high spatial and temporal resolution as well as high voltage sensitivity. It should be simple to use to reduce cycle time and costs, and preferably be nondestructive. It should also be able to perform measurements on an active integrated circuit without disturbing the circuit's operation. One instrument being explored for this application is based on a scanning probe microscopy technique called electrostatic force microscopy (EFM).

1.2 Project goals and scope

The original objective of this thesis was to measure signal skew at internal nodes of integrated circuits using a new EFM signal measurement method. However, it was soon realized that the development of electrostatic force microscopy at the University of Manitoba was limited by the lack of an EFM controller. At that time, the hardware of an existing scanning probe microscope controller was being upgraded for increased flexibility by A. Lemus [1]. Given changes to its software, this control system could be used with an EFM. Therefore, upgrading the controller's software was undertaken as a secondary objective, focusing on EFM control.

1.3 Organization of this report

This chapter establishes the motivation for measuring signals at internal nodes of integrated circuits. It states the goals of this project and describes the organization of this report.

The next chapter reviews several techniques for measuring internal node signals other than EFM, to provide the context for this work.

The following chapter briefly overviews scanning probe microscopy (SPM) techniques and technology. This is useful for understanding both electrostatic force microscopy and the SPM control system.

Next, the principles behind electrostatic force microscopy are presented. Selected modes of operation and the instrument's performance are examined.

The next two chapters describe the changes made to the scanning probe microscope controller and its user interface.

This is followed by an explanation of the new signal measurement technique, its advantages and disadvantages. Some measurements are performed using this technique and the results are discussed.

Finally, some conclusions based on the results of this project are given.

2

Internal node signal measurement techniques

2.1 Introduction

Besides the electrostatic force microscope, a number of instruments can monitor signals inside integrated circuits. A few are commercial tools, but most exist only in research laboratories and are being developed in competition with EFM. Several of these internal node signal measurement schemes are presented here to establish the context for this report.

2.2 Contact probing

The simplest way to measure a signal at an internal node of an integrated circuit is to electrically contact an interconnect with a mechanical probe and observe the signal on an oscilloscope. There are various problems with this technique that depend on the type of probe used. These problems include limited bandwidth, the necessity of removing the IC's

protective passivation layer, and the risk of damaging the IC at the contact point. Accurate positioning is difficult on finely detailed circuits, and capacitive loading of the circuit by the probe can affect its behaviour. Some probes require special coplanar waveguide contacts to be fabricated onto the IC in anticipation of internal probing.

2.3 Electron beam testing and photoemission probing

The electron beam tester (EBT) [2] is a commercial instrument based on the scanning electron microscope. It stimulates emission of secondary electrons from an integrated circuit using a focused electron beam. The energy of these secondary electrons depends on the electric potential at the focal point, and is measured with a secondary electron collector and retarding grid spectrometer. Absolute potential measurements require calibration and can be difficult, but voltage contrast measurements are often adequate for digital signals. The EBT technique is illustrated in Figure 2.1.

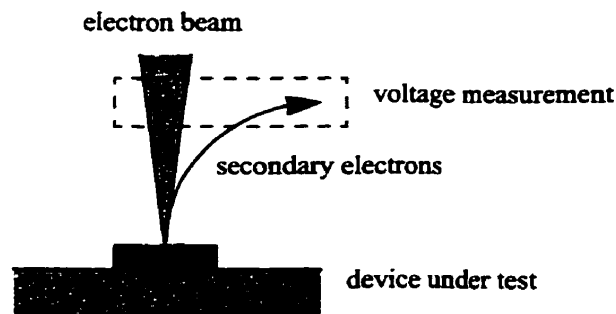


Figure 2.1 Electron beam testing.

High frequency signals with a known repetition rate can be measured with short electron pulses synchronized to the on-chip signal. These pulses sample the signal at one point in its period, just as a strobe light can reveal the position of a rotating fan when it

flashes. By shifting the phase between the pulsed electron beam and the on-chip signal, the entire high frequency waveform can be sampled. If the electron beam is simultaneously scanned across several interconnects, multiple signals can be measured at the same time. This measurement is a typical EBT technique called logic state mapping. It generates a voltage image as a function of position and phase as shown in Figure 2.2.

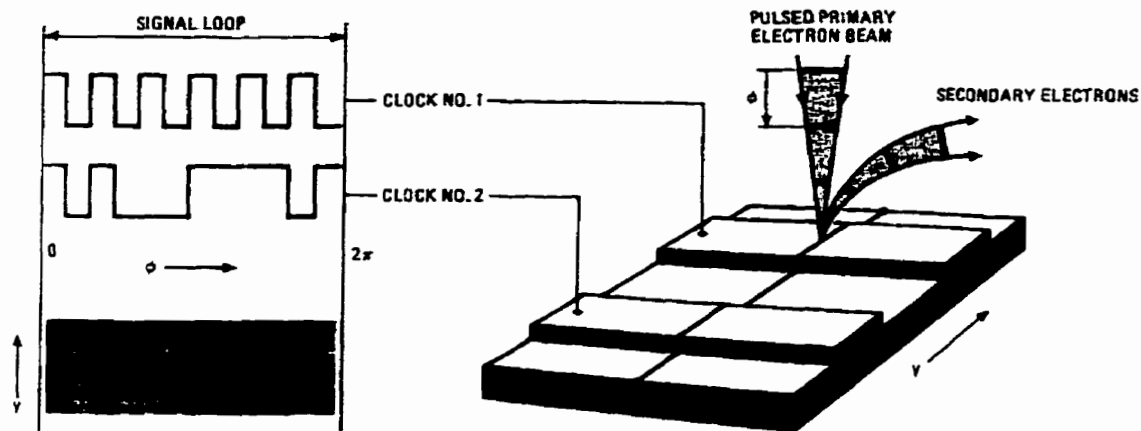


Figure 2.2 Logic state mapping in EBT [3].

Electron beam testing has a few disadvantages. Spatial and temporal resolution are limited by the focusing and chopping of the electron beam. Charge storage in the passivation layer can affect measurements, and there is a small risk of radiation damage. Also, the device under test must be placed in a vacuum chamber, which complicates and slows the testing procedure.

Photoemission probing [4] is similar to EBT, except that a laser is used to stimulate emission of secondary electrons.

2.4 Electro-optic probing

In the presence of an electric field, the index of refraction of certain crystals changes depending on the field's magnitude and direction. This change alters the polarization of a beam of light passing through such a crystal. Electro-optic probing [5] uses this effect to measure electric fields inside integrated circuits and thus infer the electric potentials of nearby conductors.

If an IC is fabricated with an electro-optic substrate such as gallium arsenide or indium phosphide, electro-optic probing can be performed simply by passing an optical probe beam through part of the substrate. Alternatively, a small electro-optic crystal can be placed at the test point for external probing, as in Figure 2.3.

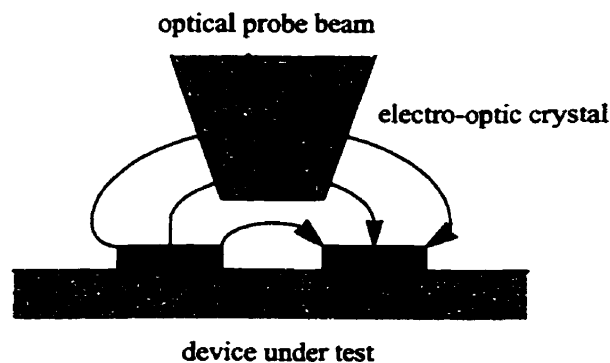


Figure 2.3 External electro-optic probing.

Electro-optic probing has excellent temporal resolution. However, its spatial resolution is limited by the optical beam width, and it can be susceptible to crosstalk effects. In the case of external probing, the crystal can capacitively load the circuit.

2.5 Photoconductive sampling

Photoconductive sampling [6] uses a small probe containing a photoconductive switch. One side of the switch is connected to the device under test, while the other side is connected to the measurement electronics. By exposing the switch to an ultrashort laser pulse, it is briefly turned on and samples the circuit signal. As in electron beam testing, by shifting the phase between the sampling pulse and the circuit signal an entire repetitive waveform can be sampled. A diagram of the photoconductive sampling setup is given in Figure 2.4.

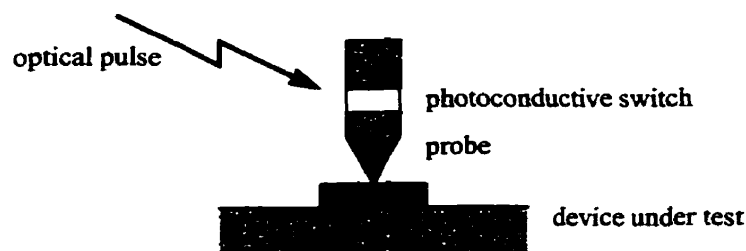


Figure 2.4 Photoconductive sampling.

The spatial resolution and capacitive loading of the photoconductive sampling technique depends on the geometry of the probe. Temporal resolution is limited by the performance of the switch.

2.6 Near-field probing

Near-field probing [7] is a simple technique using electromagnetic probes positioned close to the device under test. The circuit signal is capacitively or inductively cou-

pled to the probe for measurement. This technique has a wide bandwidth, but poor spatial resolution.

3

An overview of scanning probe microscopy

3.1 Introduction

Conventional optical and electron microscopes use lenses to refract radiation that has interacted with a sample. Diffraction limits the spatial resolution of these microscopes to about half the wavelength of the radiation used. In contrast, scanning probe microscopes (SPMs) measure localized properties of a sample using its interaction with a very sharp probe operated very close to its surface. Scanning probe microscopes are not limited by diffraction and can achieve much greater spatial resolution. Developed in the early 1980s, SPMs were quickly adopted in many fields and have been used to study everything from magnetic domains to DNA. This chapter briefly introduces the field of scanning probe microscopy, with emphasis on scanning force microscopy.

3.2 Scanning tunnelling microscopy

In 1982, Binnig *et al.* published a paper describing an instrument capable of measuring surface topography with atomic resolution [8]. Gerd Binnig and Heinrich Rohrer would later receive the Nobel Prize for their invention of the scanning tunnelling microscope (STM).

The STM was the first scanning probe microscope. As shown in Figure 3.1, it uses a sharp, conducting probe placed less than 1 nm from the surface of a conducting sample. When a bias voltage is applied between the probe and sample, the short distance between them allows charge carriers to tunnel from one to the other. Since the tunnelling current decays exponentially with distance, the gap between the closest points on the probe and the sample can be measured with tremendous resolution. By scanning the probe over the sample, it is possible to image individual atoms on the sample's surface.

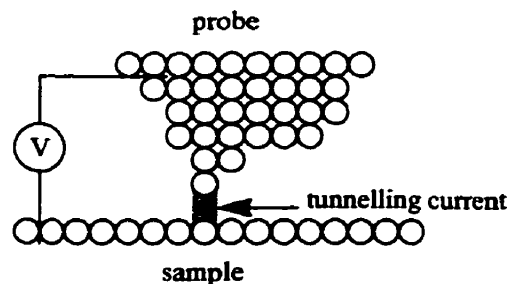


Figure 3.1 The probe and sample of a scanning tunnelling microscope.

Scanning tunnelling microscopes can be operated in two modes: constant height and constant current.

In constant height mode, the probe is moved in a flat plane above the sample. The variations in the tunnelling current versus position form an image.

In constant current mode, the tunnelling current is kept constant by moving the probe closer or further from the sample while scanning. For example, if an increase in the tunnelling current is detected, the probe is pulled away from the sample to compensate. In this mode, the variations in the probe height constitute the image. Constant height mode and constant current mode are compared in Figure 3.2.

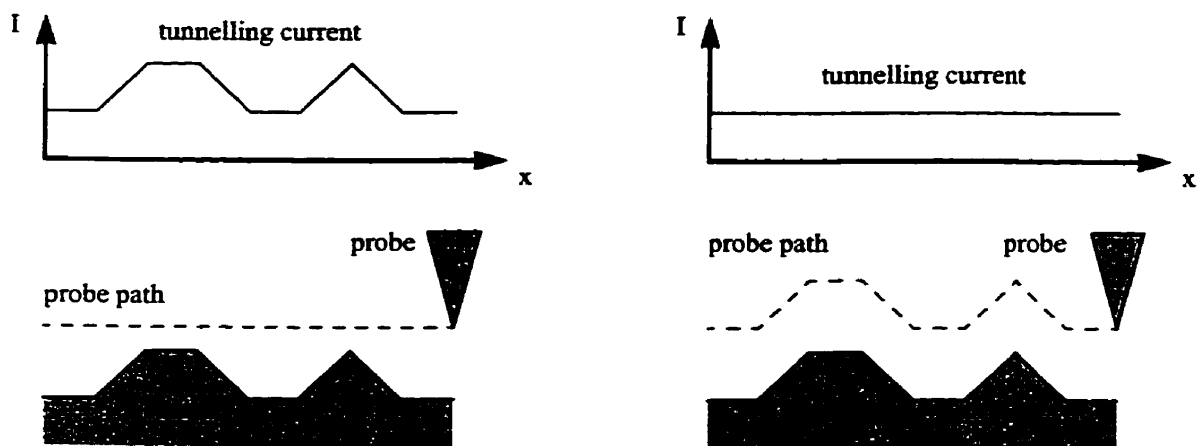


Figure 3.2 Constant height mode (left) versus constant current mode in STM.

Strictly speaking, the images obtained with a scanning tunnelling microscope are tunnelling probability images, not distance images. Both the surface and probe topographies and their electronic structures contribute to the images. Although this coupling is problematic for those interested in mapping the surface topography, it has been exploited by some to perform localized spectroscopy measurements.

3.3 Scanning force microscopy

Scanning force microscopy (SFM) is a broad subclass of scanning probe microscopy. As shown in Figure 3.3, a scanning force microscope uses a sharp probe mounted on

the end of a small, flexible cantilever beam. Forces acting between the probe and sample deflect the cantilever according to Hooke's Law. These forces typically decay as the probe-sample gap increases. By scanning the probe over the sample and mapping the interaction force, localized surface properties can be measured.



Figure 3.3 The probe and sample of a scanning force microscope.

Like the STM, SFMs have two modes of operation: constant height mode and constant force mode.

Constant height mode SFM is much like constant height mode STM. The probe is scanned in a flat plane over the sample and the cantilever deflections are used to map the surface forces.

In constant force mode, a feedback control system modulates the probe-sample separation to maintain a constant cantilever deflection and thus a constant force. The movement of the sample forms a constant force contour, which is recorded as an image. This mode is slower than constant height mode, but is more popular because it limits the probe-sample force.

In addition to these two operating modes, scanning force microscopes have three different imaging modes: contact imaging, noncontact imaging, and intermittent contact imaging. The choice of imaging mode depends on what force is to be measured.

3.3.1 Contact imaging

The first scanning force microscope was the contact mode atomic force microscope (AFM), described by Binnig, Quate, and Gerber in 1986 [9]. This microscope uses a probe that makes a gentle physical contact with a sample, like the needle of a record player contacting a record. By scanning the probe over the surface, the repulsive interatomic forces between them can be mapped. This map follows the surface topography, with spatial resolution limited by the sharpness of the probe. Although the AFM does not boast the same resolution as the STM, it has the advantage that the probe and sample do not have to be conductors.

The contact imaging mode has been extended to measure various sample properties in addition to topography. Scanning capacitance microscopy (SCaM) [10] uses a capacitance sensor to measure variations in probe-sample capacitance as the probe is scanned over the sample. Scanning resistance microscopy (SRM) [11] uses the current flowing through the contact between a biased, conducting probe and the sample to measure localized sample resistance.

3.3.2 Noncontact imaging

The noncontact imaging mode uses a larger probe-sample gap and longer ranging forces than the interatomic forces of contact mode to image a sample. The long range nature of these forces gives this imaging mode poorer resolution than contact mode, but it is useful when contact between the probe and sample is undesirable. For example, non-contact mode AFM using attractive van der Waals forces can image soft samples that would be deformed by the contact pressure of the probe in contact mode.

Noncontact SFMs commonly use a technique called force gradient detection, or *ac* detection, to measure the cantilever deflection. Since the forces acting on the probe in noncontact mode are typically much smaller than those in contact mode, the cantilever deflection is much smaller and detection of these forces is more difficult. Force gradient detection is more sensitive than simple force detection.

Force gradient detection exploits the behaviour of the cantilever versus the derivative of the probe-sample force with respect to the distance between them [12]. This derivative is called the force gradient. A cantilever's resonant frequency varies with the square root of its effective spring constant, which in turn varies with the force gradient it experiences. An SFM can thus be operated in constant force mode by adjusting the probe-sample distance to keep the cantilever resonant frequency constant. Measuring the resonant frequency usually requires vibrating the cantilever at or near that frequency.

The noncontact imaging mode is used, as previously mentioned, in noncontact AFM to image soft samples. It has also been used to study charge distributions with electrostatic force microscopy [13], and to image magnetic domains with magnetic force microscopy (MFM) [14].

3.3.3 Intermittent contact imaging

Intermittent contact imaging, or tapping mode, is similar to noncontact AFM. In this case, the probe is allowed to gently touch the sample while it is vibrating. This technique allows AFM imaging with high resolution but low force.

3.4 Scanning force microscope construction

This section describes the main components of a typical scanning force microscope. A schematic diagram of such a system is given in Figure 3.4.

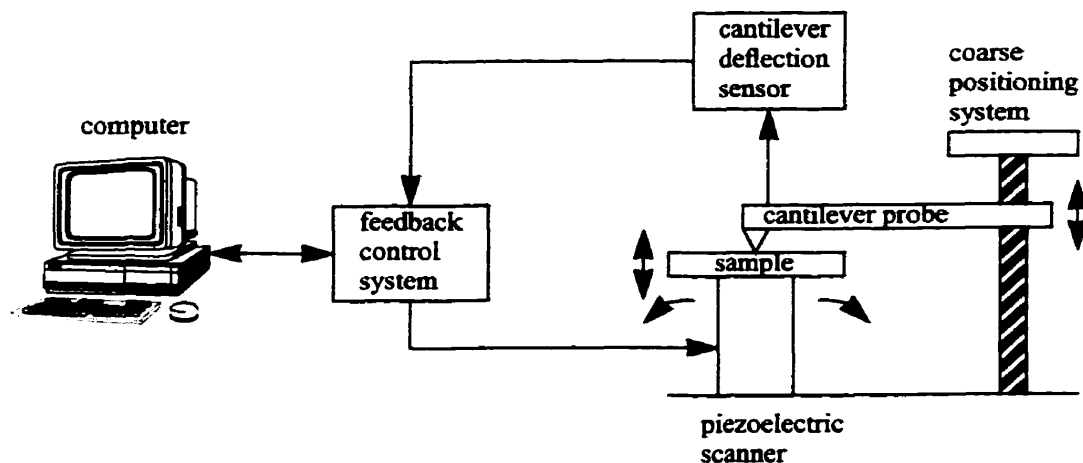


Figure 3.4 Schematic diagram of a typical scanning force microscope.

3.4.1 Cantilever probe

Commercial SFM cantilevers are usually microfabricated from silicon or silicon nitride, with a pyramidal or conical probe at the end. Typical dimensions for the cantilever might be $100\text{ }\mu\text{m}$ in length and $1\text{ }\mu\text{m}$ in thickness, with a resonant frequency on the order of 10 kHz and a spring constant of 1 N/m . The probes of magnetic force microscope cantilevers are coated with magnetic material, while electrostatic force microscope cantilevers are coated with enough metal to form a conducting path to the probe. Figure 3.5 shows a common, V-shaped cantilever. The V-shape makes the cantilever resistant to twisting, which is generally undesirable. “Diving-board” shaped cantilevers and macroscopic cantilevers are also used in some applications.

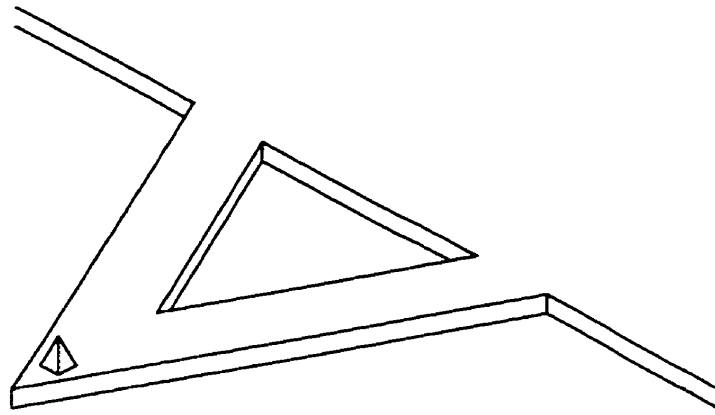


Figure 3.5 Diagram of a microfabricated SFM cantilever, with probe pointing upwards.

The mechanical frequency response of an SFM cantilever to external forces is important when studying a noncontact technique. For $Q \gg 1$ the magnitude of this response is approximately

$$|G(\omega)| = \frac{Q/k}{\sqrt{1 + Q^2 (\omega^2/\omega_r^2 - 1)^2}} \quad (3.1)$$

where ω_r is the cantilever resonant frequency, Q is the quality factor, and k is the spring constant [15]. Far below the resonant frequency, the cantilever deflects according to Hooke's Law. Near that frequency, the deflection of the cantilever is enhanced by its quality factor. Above resonance, the cantilever does not respond to applied forces.

3.4.2 Piezoelectric scanner

The mechanism used in many SFMs to scan the probe relative to the sample is a cylinder of piezoelectric material such as lead zirconate titanate (PZT). The sample is mounted atop the cylinder, and the sample is scanned while the probe remains stationary.

Piezoelectric materials change their dimensions under an applied electric field. By attaching four electrodes to the outside of a hollow piezoelectric tube and one to the inside

as shown in Figure 3.6, it is possible to distort the tube and move a sample mounted atop it in three dimensions by applying the appropriate voltages to the electrodes. In constant force mode, these voltages correspond to the probe-sample relative position and are used to form the image. Piezoelectric tube scanners have ranges up to about 10 μm in the axial direction and about 100 μm in the transverse plane. Applied voltages are typically hundreds of volts.

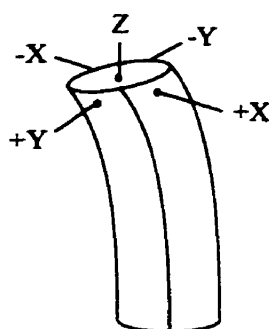


Figure 3.6 Diagram of a piezoelectric tube scanner.

Piezoelectric scanners are used because they allow subnanometre resolution and respond quickly to applied voltage. However, they are subject to a variety of nonlinearities such as hysteresis, creep, aging, and coupling between axes. These nonlinearities can be compensated for with software correction algorithms or feedback control systems.

3.4.3 Cantilever deflection sensor

In the original AFM, the deflection of the cantilever was detected with an STM probe piggybacked onto the AFM cantilever. Measuring the tunnelling current between a fixed STM probe and the AFM cantilever gives excellent resolution, but is sensitive to the surface conditions of the cantilever.

The most common deflection detection system used in modern SFMs is the “beam-bounce” system shown in Figure 3.7. A laser beam is reflected off the back of the cantilever onto a split photodetector. As the cantilever bends, the reflected beam moves from one half of the photodetector to the other. The difference in power incident on the two halves of the photodetector is used to measure the cantilever deflection. This system is simple, reliable, and gives subnanometre resolution [16].

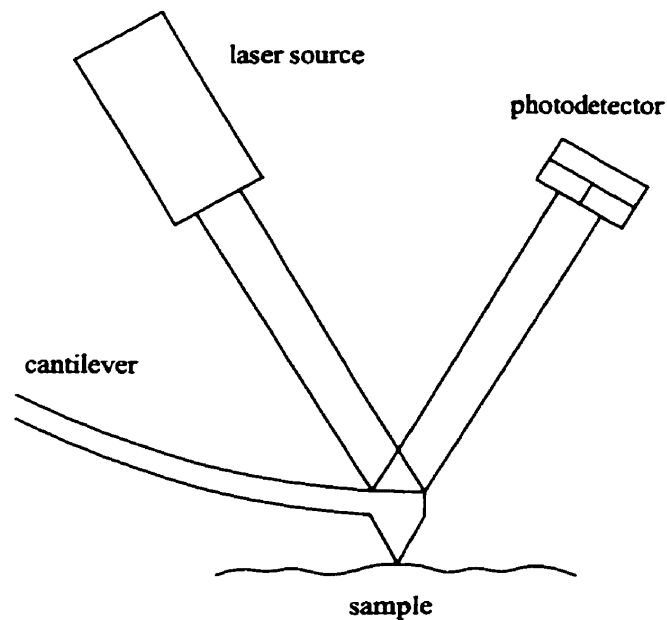


Figure 3.7 The “beam-bounce” cantilever deflection detection scheme.

Other techniques that have been used to detect the SFM cantilever deflection include capacitive detection, optical interferometry, and piezoresistive cantilevers.

3.4.4 Computers and electronic controls

Scanning probe microscopes would not be practical industrial tools without the aid of computers and electronic controls. Electronic control systems can perform delicate operations such as maintaining a constant cantilever deflection while scanning a contact

mode AFM probe across a rough surface. Computers typically provide a simple user interface to these control systems, and are used to record and analyse data. The roles of these two components are discussed in greater detail in later chapters.

3.5 Other techniques

Besides STM and SFM, researchers have developed a variety of scanning probe microscopes to investigate different surface properties. Scanning near-field optical microscopy (SNOM) pushes the spatial resolution of optical microscopy beyond the diffraction limit. Scanning electrochemical microscopy (SECM) uses Faradaic current to measure electrochemical properties similar to the way the STM uses tunnelling current to measure electronic structure. Scanning probe microscopes have also physically and chemically modified surfaces.

4

Signal measurement by electrostatic force microscopy

4.1 Introduction

Electrostatic force microscopy is a noncontact SFM technique that has imaged surface charges on insulators [13], contact potential differences [17], and dc electric potential distributions [18]. In this work, it is used to measure signals inside integrated circuits.

Signal measurement inside integrated circuits by electrostatic force microscopy is somewhat different from other SPM techniques, because the property being measured changes rapidly with time. The EFM often simply monitors a signal at a single point, and does not scan over the sample at all. This chapter explains the principles behind electric potential measurement by EFM. It presents some signal measurement techniques, and examines the instrument's performance.

4.2 Theory

The force of interest when measuring electric potentials by EFM is the electrostatic force acting on a conducting probe of known potential due to a second conductor of unknown potential. Figure 4.1 depicts an EFM probe above a single integrated circuit interconnect.

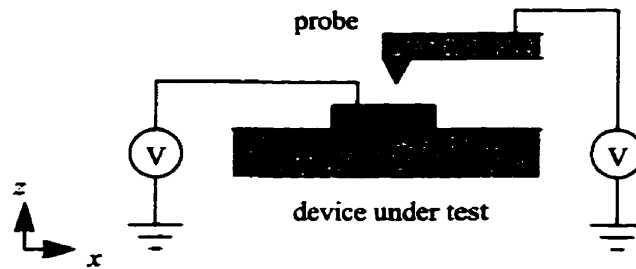


Figure 4.1 An EFM probe above a single integrated circuit interconnect.

An expression for the electrostatic force exerted on the probe can be derived as follows. The electric potential energy of the system in Figure 4.1 is given by

$$U = \frac{1}{2}CV^2 \quad (4.1)$$

where C is the capacitance between the probe and interconnect and V is the potential difference between them.

Assuming the potentials of the probe and interconnect are fixed, the electrostatic force experienced by the probe is

$$\mathbf{F} = \nabla U = \nabla \left(\frac{1}{2}CV^2 \right) = \frac{1}{2}(\nabla C)V^2. \quad (4.2)$$

The cantilever deflection sensor in the EFM is most sensitive to motion in the vertical direction. The force in this direction is

$$F_z = \frac{1}{2} \frac{\partial C}{\partial z} V^2. \quad (4.3)$$

If the capacitance gradient $\frac{\partial C}{\partial z}$ is kept constant, then the force acting on the probe will be proportional to the square of the potential difference between it and the interconnect. The capacitance gradient varies with the relative positions and orientations of the two conductors and the dielectric medium between them.

4.3 Probe signal amplitude modulation

The signals of interest inside an integrated circuit usually change much too quickly for the EFM cantilever to respond. However, if the circuit signal has a known repetition rate, the beating between it and a probe signal slightly offset in frequency can be used to alias the circuit signal to a much lower frequency [19]. This technique is similar to the sampling of high frequency signals in electron beam testing. Unfortunately, it requires the frequency difference between the two signals to be well below the resonant frequency of the cantilever, which is difficult and expensive at high frequencies.

The favoured technique at the University of Manitoba is to modulate the amplitude of a probe signal that is synchronized with the circuit signal [20]. By choosing the modulation frequency to be near or below the cantilever's resonant frequency, part of the generated force also falls within this range and the cantilever is able to respond. To understand this technique, we should first rewrite (4.3) as

$$F_z(x, y, z, t) = \frac{1}{2} \frac{\partial}{\partial z} C(x, y, z) [v_p(t) - v_c(x, y, t) + \phi]^2 \quad (4.4)$$

where $\frac{\partial}{\partial z}C(x, y, z)$ is the capacitance gradient between the probe at point (x, y) and the part of the circuit being probed, $v_p(t)$ is the probe potential, $v_c(x, y, t)$ is the circuit potential, and ϕ represents the effects of surface and trapped charge and work function differences.

Now, let the probe potential be an amplitude modulated sampling signal as follows:

$$v_p(t) = [A + K \cos \omega_m t] v_s(t) \quad (4.5)$$

where A and K are dimensionless modulation parameters, ω_m is the modulation frequency, and $v_s(t)$ is the sampling signal. Both $v_c(x, y, t)$ and $v_s(t)$ have period $T = \frac{2\pi}{\omega_0}$ where $\omega_0 \gg \omega_m$.

If we now substitute and expand (4.4), we obtain

$$\begin{aligned} F_z(x, y, z, t) = & \frac{1}{2} \frac{\partial}{\partial z} C(x, y, z) [A^2 v_s^2(t) + K^2 v_s^2(t) \cos^2 \omega_m t + v_c^2(x, y, t) + \phi^2 \\ & + 2AK v_s^2(t) \cos \omega_m t - 2A v_s(t) v_c(x, y, t) + 2A v_s(t) \phi - 2K v_s(t) v_c(x, y, t) \cos \omega_m t \\ & + 2K v_s(t) \phi \cos \omega_m t - 2v_c(x, y, t) \phi] \end{aligned} \quad (4.6)$$

This force has frequency components at dc, ω_m , $2\omega_m$, $n\omega_0$, $n\omega_0 \pm \omega_m$, and $n\omega_0 \pm 2\omega_m$. We are interested only in the component at ω_m , which can be written as

$$\begin{aligned} F_z(x, y, z, t) \Big|_{\omega = \omega_m} & \\ = & \frac{K}{T} \frac{\partial}{\partial z} C(x, y, z) \left\{ \int_0^T [A v_s^2(t) - v_s(t) v_c(x, y, t) + v_s(t) \phi] dt \right\} \cos \omega_m t \end{aligned} \quad (4.7)$$

where $\frac{1}{T} \int_0^T f(t) dt$ is the mean of any function $f(t)$ with period T .

If we choose ω_m to be the cantilever resonant frequency ω_r , the resulting deflection at ω_r will be mechanically amplified by the cantilever's quality factor and improve the instrument's signal-to-noise ratio. From (3.1), this deflection is

$$\begin{aligned} \Delta z(x, y, z, t) \big|_{\omega = \omega_r} \\ \approx \frac{QK}{kT} \frac{\partial}{\partial z} C(x, y, z) \left\{ \int_0^T [A v_s^2(t) - v_s(t) v_c(x, y, t) + v_s(t) \phi] dt \right\} \cos \omega_r t \end{aligned} \quad (4.8)$$

where Q is the cantilever quality factor and k is the cantilever spring constant.

There are two possible ways to deduce $v_c(x, y, t)$ from the cantilever deflection. By carefully choosing A and $v_s(t)$, the force at ω_r can be made to vanish. This force nulling scheme has the advantage that the poorly conditioned factors Q , k , and $\frac{\partial}{\partial z} C(x, y, z)$ do not need to be known. However, it requires some form of feedback control or manual intervention. Neglecting ϕ , $v_c(x, y, t)$ can then be determined absolutely from the choice of A and $v_s(t)$. Alternatively, the cantilever deflection due to a fixed A and $v_s(t)$ can simply be measured. This voltage contrast scheme is simpler, but depends on the values of Q , k , and $\frac{\partial}{\partial z} C(x, y, z)$. Relative measurements are still possible if these factors remain constant. Examples of both of these signal measurement schemes are given in the next two sections.

A block diagram of the amplitude modulation technique is given in Figure 4.2. The deflection at ω_r is measured with a lock-in amplifier.

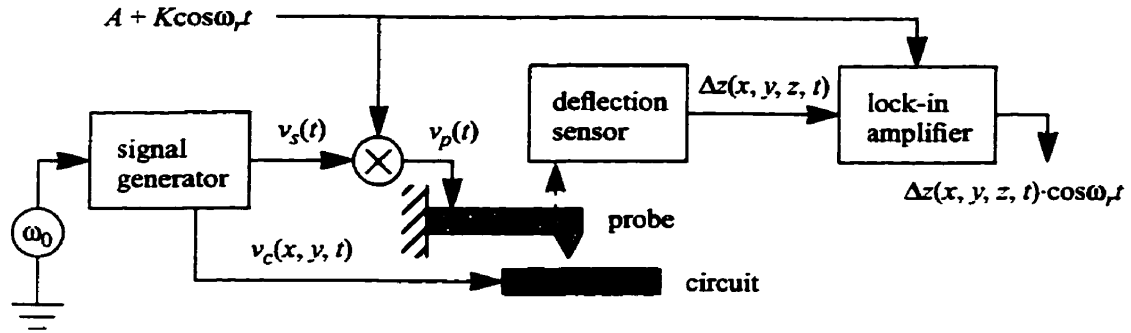


Figure 4.2 Block diagram of the amplitude modulated EFM potential measurement technique.

In practice, the modulating signal $A + K\cos\omega_r t$ is often replaced by a 50% duty cycle square wave with extrema of 0 and 1, because it is easy to implement via a multiplexer or switch. Since only the first harmonic of the cantilever deflection is measured, this is equivalent to setting $A = \frac{1}{2}$ and $K = \frac{2}{\pi}$, the first two Fourier coefficients of this square wave.

4.4 Sinusoidal sampling

Suppose the circuit signal and the sampling signal are sinusoids as follows:

$$v_c(x, y, t) = V_c \sin(\omega_0 t + \phi_c) \quad (4.9)$$

$$v_s(t) = V_s \sin(\omega_0 t + \phi_s). \quad (4.10)$$

Then (4.8) reduces to

$$\Delta z(x, y, z, t)|_{\omega=\omega_r} \approx \frac{QK}{2k} \frac{\partial}{\partial z} C(x, y, z) [AV_s^2 - V_s V_c \cos(\phi_s - \phi_c)] \cos\omega_r t. \quad (4.11)$$

Notice that the term involving ϕ has vanished. Also, if $AV_s = V_c$ and $\phi_s = \phi_c$, then the force at ω_r vanishes regardless of the values of Q , k , and $\frac{\partial}{\partial z}C(x, y, z)$. The circuit signal magnitude and phase can thus be determined absolutely by adjusting A , V_s , and ϕ_s until the deflection at ω_r is nulled. This vector measurement through force nulling is useful for characterizing sinusoidal signals at individual points inside an integrated circuit.

4.5 Pulse sampling

Next, suppose the sampling signal is a train of narrow, rectangular pulses of width δ and height V_s

$$v_s(t) = \sum_{n=-\infty}^{\infty} V_s G_{\delta}(t - nT - t_0) \quad (4.12)$$

where t_0 is a controllable delay and $\delta \ll T$. This is shown in Figure 4.3.

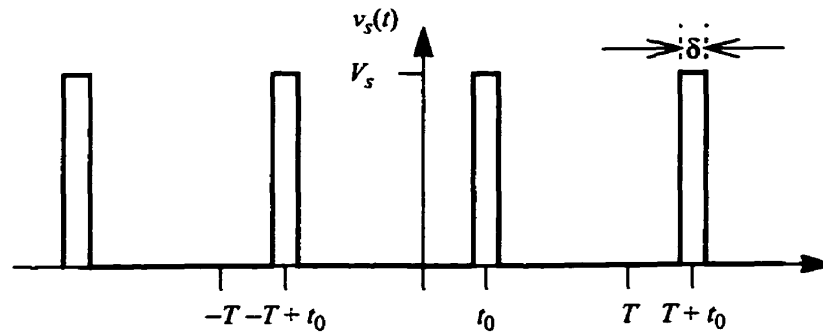


Figure 4.3 A rectangular pulse train.

Substituting into (4.8) and realizing that $\int_0^T v_s(t) v_c(x, y, t) dt \approx V_s \delta v_c(x, y, t_0)$ if δ is small, the deflection becomes

$$\Delta z(x, y, z, t)|_{\omega = \omega_r} \approx \frac{QKV_s \delta}{kT} \frac{\partial}{\partial z} C(x, y, z) [AV_s - v_c(x, y, t_0) + \phi] \cos \omega_r t. \quad (4.13)$$

In this case ϕ does not vanish, so $v_c(x, y, t)$ cannot be determined exactly by nulling the force without additional calibration. However, if A and V_s are kept constant and t_0 is swept from 0 to T , the shape of an arbitrary repetitive waveform $v_c(x_0, y_0, t)$ can be determined at a fixed point (x_0, y_0) inside an integrated circuit by measuring the magnitude of the cantilever vibration at ω_r . Alternatively, t_0 may be fixed and the probe scanned over the sample to obtain a snapshot of the relative potentials $v_c(x, y, t_0)$ on several interconnects at time t_0 . The results of this pulse sampling voltage contrast technique are similar to those of the logic state mapping technique used in electron beam testing.

4.6 Performance

4.6.1 Spatial resolution

At the beginning of this chapter, we assumed that a single integrated circuit interconnect generates the electrostatic force acting on the EFM probe. However, a real IC contains many interconnects. The net electrostatic force acting on an EFM probe is the sum of the forces generated by all of the nearby conductors. Thus, other interconnects near to one being probed can introduce crosstalk effects into the measurement. It is important to characterize the extent of these effects to better interpret EFM measurements.

This delocalization of the interaction between an EFM probe and sample is described by the instrument's spatial resolution, an important characteristic of any microscope. Since the contribution of a given interconnect to the net electrostatic force acting on an EFM probe depends on their electric potentials and geometries, it is difficult to define spatial resolution for all probes, circuits, and flying heights. However, specific cases can establish trends and provide a basis for estimating the spatial resolution in other cases. For example, Said [15] calculated the force between a macroscopic probe and an infinite ground plane at various flying heights, and defined spatial resolution as the radius of a circular area that induces half of the total electrostatic force on the probe. With a potential difference of 1 V, the spatial resolution varied from 700 nm at 100 nm flying height to 15 μm at 1 μm flying height. Similarly, Mueller *et al.* [21] modelled a micromachined cantilever probe over a coplanar waveguide and defined spatial resolution as the diameter at which the electrostatic force on the probe drops to $1/e$ of its maximum. With a potential difference of 1 V, the spatial resolution was 300 nm at a flying height of 100 nm. In both cases, the spatial resolution deteriorated rapidly as the flying height increased.

4.6.2 Temporal resolution

Temporal resolution is relevant when measuring a time-varying signal, as in the pulse sampling technique. It can be divided into two characteristics: time resolution and phase resolution.

Time resolution is the error present in transition time measurements. Sampling a signal with pulses of nonzero width averages the signal over that time, essentially low-pass filtering it. Fast events and sharp edges become blurred. The time resolution of the

EFM is limited by the minimum width of the sampling pulses that can be generated. Narrower pulses require more expensive equipment with greater bandwidth, capable of producing shorter rise- and fall-times. Also, from (4.13), reducing δ while keeping everything else constant reduces the cantilever deflection, degrades the signal-to-noise ratio, and makes measurements more difficult. The pulse amplitude can be increased to compensate, but this requires equipment capable of generating even faster transitions. At the University of Manitoba, pulses of 100 ps width have been achieved.

Phase resolution is the error present in propagation delay and timing measurements. In theory, this is limited by the resolution with which the sampling pulses can be shifted in time, but in practice noise in the measured signals usually delocalizes edges beyond this limit. The effect of this noise becomes less significant as time resolution improves and measured signal transitions become steeper.

4.6.3 Electric potential resolution

The resolution of electric potential measurements using the EFM is defined by the noise in those measurements. The dominant noise source in EFM is the vibration of the cantilever as it exchanges thermal energy with its environment. The magnitude of this thermal noise as it appears in the measured electric potential is derived below.

According to Said [15], the mean-square thermal vibration of the cantilever is

$$\overline{z_{th}^2} = \frac{4kk_B T_a B}{2\pi Q\omega_r} \int_0^\infty |G(\omega)|^2 d\omega \quad (4.14)$$

where k is the cantilever spring constant, k_B is Boltzmann's constant, T_a is the ambient temperature, B is the measurement bandwidth, Q is the cantilever quality factor, ω_r is the

cantilever resonant frequency, and $G(\omega)$ is the cantilever mechanical frequency response. Similarly, the mean-square cantilever deflection due to an electrostatic force is

$$\overline{z^2} = \frac{\overline{F_z^2(\omega)}}{2\pi} \int_0^\infty |G(\omega)|^2 d\omega \quad (4.15)$$

where $\overline{F_z^2(\omega)}$ is the power spectral density of the electrostatic force. Equating (4.14) and (4.15) and taking square roots, the root-mean-square (rms) electrostatic force that would generate the same cantilever deflection as the thermal noise is

$$\sqrt{\overline{F_z^2(\omega)}} = \sqrt{\frac{4kk_B T_a B}{Q\omega_r}}. \quad (4.16)$$

The rms cantilever deflection due to an electrostatic force can also be expressed as

$$\sqrt{\overline{F_z^2(\omega)}} = \frac{1}{2} \frac{\partial}{\partial z} C(x, y, z) V_{rms}(\omega) \quad (4.17)$$

where $V_{rms}(\omega)$ is the rms electric potential difference between the probe and the device under test. Combining (4.16) and (4.17), the minimum detectable rms electric potential difference is

$$V_{rms} = \frac{2}{\frac{\partial}{\partial z} C(x, y, z)} \sqrt{\frac{4kk_B T_a B}{Q\omega_r}}. \quad (4.18)$$

For a micromachined cantilever positioned $1 \mu\text{m}$ above a $3 \mu\text{m}$ wide interconnect, Noruttun [22] estimated this noise at $1 \text{ mV}/\sqrt{\text{Hz}}$.

4.6.4 Invasiveness

An ideal instrument does not affect the quantity it is attempting to measure. However, all real instruments are invasive to some degree; that is, performing a measurement alters the quantity being measured.

A signal probing device like the EFM can disturb an integrated circuit by introducing an additional load to the circuit. Since EFM is a noncontact technique, resistive loading is not an issue. However, capacitive loading is possible as the cantilever probe acts as a shunt capacitor. If this capacitance is large enough, it can act as a lowpass filter and increase signal rise- and fall-times. The probe may even inject its signal into the circuit.

Bridges and Thomson [23] estimated the coupling capacitance between a 3 μm wide interconnect and a micromachined cantilever 1 μm above it at 230 aF. This value is small, corresponding to an impedance of 7 k Ω at 100 GHz. This load is unlikely to affect most circuits, although coupling will increase as the flying height decreases.

5

Changes to the digital SPM controller software

5.1 Introduction

The objective of the work described in this chapter was to adapt the core software of an existing digital SPM controller for use with all of the microscopes in the University of Manitoba's SPM laboratory. At the start of this project, this consisted of an STM, AFM, SRM and EFM.

The primary function of an SPM controller is to control the relative positions of the probe and sample. The controller usually acts as a compensator in a feedback control system that maintains a constant current in constant current mode STM, or constant force in constant force mode SFM. Figure 5.1 shows an example of a feedback control system for constant force mode SFM.

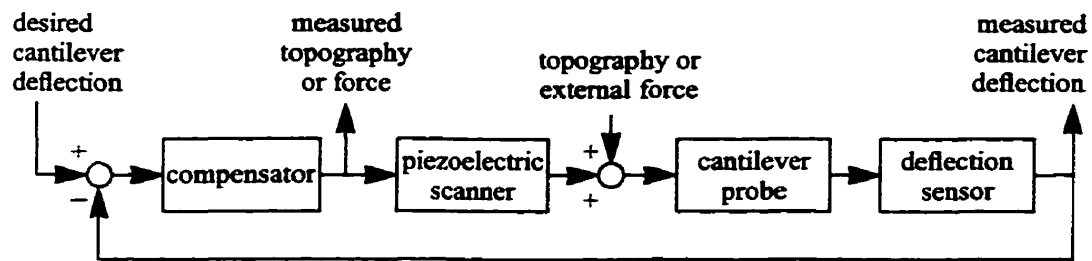


Figure 5.1 Block diagram of a constant force mode SFM feedback control system.

The digital SPM controller was originally designed for use with the STM and AFM, so only a few changes were needed to account for some recently upgraded hardware before they could be used.

The SRM is essentially a contact mode AFM that also uses the current flowing through a biased, conducting probe to measure the local sample resistance. Since the cantilever deflection and probe current are unrelated, the SRM can measure both sample resistance and sample topography simultaneously. An SRM controller must record both sets of data.

The EFM is a noncontact mode SFM that uses the electrostatic force between a probe and sample to measure electric potentials on the sample's surface. This microscope is somewhat different from STM, AFM and SRM. The largest part of the work described in this chapter concerns the addition of a new scanning algorithm for EFM.

This chapter explains the motivation for scanning EFM and the proposed implementation. It describes the digital SPM controller hardware and software, discusses the changes in the software and presents some results.

5.2 Scanning electrostatic force microscopy

When measuring signals inside integrated circuits with EFM, it is often possible to pinpoint the desired measurement locations with an optical microscope or AFM. However, sometimes signals are not correlated with the IC's topography. In this case, the EFM probe must scan across the sample surface.

Scanning EFM would allow imaging of potential distributions inside a device, or magnitude and phase variations in a microwave circuit. It could also create images similar to the logic state mappings generated by electron beam testers.

As previously discussed, the electrostatic force detected by the EFM varies with both the probe-sample potential difference and the capacitance gradient. If the EFM simply operated as a constant force or constant height noncontact SFM, the resulting image would be a combination of the circuit geometry and electric potential. These two effects must be separated to measure either property. Although it is possible to scan the probe along a constant capacitance gradient contour above the surface [24], this method requires a second lock-in amplifier and its behaviour is unclear when the probe is far from other conductors. A better approach is needed.

A simple solution is to measure the electrostatic force at a constant flying height above the surface. Although this technique does not completely eliminate the effect of the capacitance gradient, it should at least allow comparisons of the electrostatic forces above the centres of two similar interconnects, where measurements are most important. If we ignore the force on the cantilever, for a small flying height any interconnect appears to be

a large conducting plane exerting a force on the probe. For a given dielectric medium, the capacitance gradients at the same small flying height above two different interconnects will be the same, so the electrostatic forces there should be comparable. This method also ignores topographic features such as a slight tilt added to the integrated circuit when it is mounted on the piezoelectric scanner.

Since the EFM must measure the sample topography to separate it from the electrostatic force data, it can easily record it as well.

5.3 The digital SPM controller hardware

The digital SPM controller modified in this project was originally designed by K. Yackoboski and G. McGonigal, and was intended for use with a constant force AFM or constant current STM [25]. A digital control system provides tremendous flexibility in programming and offers better noise immunity than analog systems.

The heart of the controller is a software proportional-integral (PI) compensator, running on a Motorola DSP56001 digital signal processor (DSP) at 20.48 MHz. This software senses the cantilever deflection signal or tunnelling current signal through an analog-to-digital converter (ADC), and returns signals to the microscope through a set of digital-to-analog converters (DACs). An operator can adjust scan size and location, scan frequency, STM bias voltage and compensator parameters through a graphical user interface on an attached PC. Automatic probe approach using a microscope's coarse positioning stepper motor is also possible. The images produced are 256 pixels by 256 pixels with

65 536 greyscales, and can be stored on the PC. Figure 5.2 shows a schematic diagram of the controller hardware.

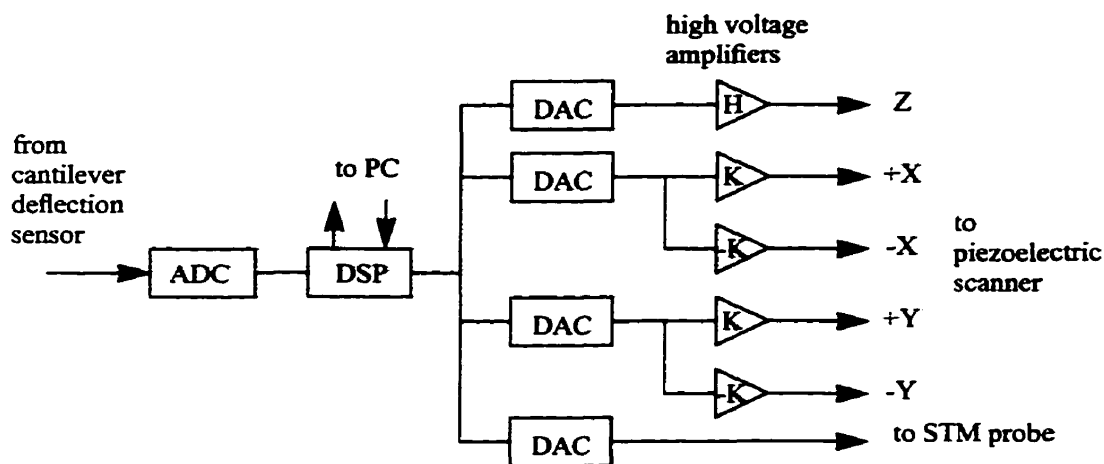


Figure 5.2 Schematic diagram of the original digital SPM controller hardware.

The controller hardware was recently modified by A. Lemus to accommodate a wider variety of SPMs [1]. The new controller features five additional ADCs and two additional DACs, all with greater dynamic range, and a digitally programmable lowpass filter at the input to one of the ADCs. The interface between the DSP and the other devices is implemented on a field programmable gate array (FPGA) to allow easy changes to the hardware. The ports accessible to the operator were also altered to simplify the attachment of several different microscopes. Figure 5.3 shows a schematic diagram of the revised controller hardware.

5.4 The digital SPM controller software

The SPM controller software is essentially a set of interrupt service routines (ISRs) written in C and inline assembly. After setting up global variables for storing measure-

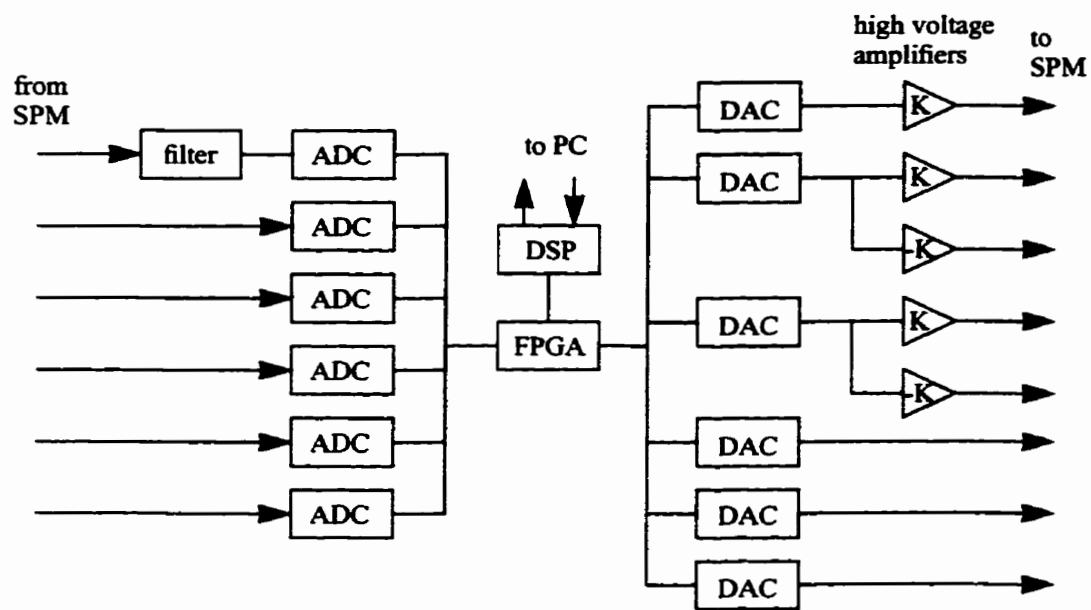


Figure 5.3 Schematic diagram of the new SPM controller hardware.

ments and the state of the system, the DSP software enters an infinite loop. An internal timer regularly calls an ISR that performs PI compensation, data acquisition, and updates the two counters that control the piezoelectric scanner's lateral position. The rate at which this ISR is called is a fraction of the DSP's clock speed. At the end of each scan line, the ISR interrupts the PC to start the transfer of the data for that line. Other ISRs allow the PC to change parameters used by the compensator ISR.

The compensator ISR performs the following steps. The terminology used in this discussion assumes that a constant force AFM is attached to the controller.

1. Calculate the difference between the desired cantilever deflection and the actual deflection.
2. Perform proportional-integral compensation on this difference, clip the result if it exceeds the DAC range, and send it to the z- (axial) electrode of the piezoelectric scanner.

3. Update the x - (fast transverse axis) position.
4. Update the feedback counter. This counter allows the ISR to perform several compensation cycles between image points, if the scan frequency allows. This relationship is discussed further at the end of this section.
5. If this is an image point, record the value sent to the z -electrode DAC and reset the feedback counter. There are 256 image points per line.
6. If this is not the start or end of a line, exit the ISR.
7. If this is the start of a line, change the x -axis direction and exit.
8. If this is the end of a line, change the x -axis direction.
9. If this is the first or last line in the image, change the y - (slow transverse axis) direction. There are 256 lines per image.
10. Update the y -position.
11. Begin transferring the recorded z -electrode values for this line to the PC, and exit. Calls to the compensator ISR are disabled during data transfer.

The algorithm performed by the ISR can be described more concisely as follows.

1. Perform a forward pass, recording the topographical data. At the end of the line, transfer the data to the PC and go to the next line.
2. Return to the start of the line. The data acquired during this return pass will be overwritten during the next forward pass. Acquiring data in only one fast-axis direction helps avoid problems with piezoelectric

scanner hysteresis. Figure 5.4 shows the resulting raster scanning pattern.

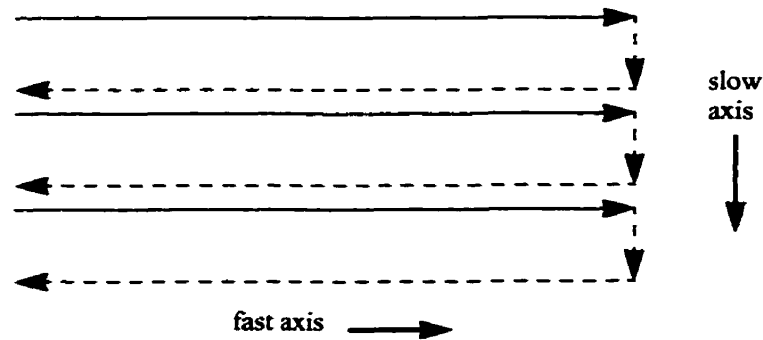


Figure 5.4 Raster scanning pattern.

The AFM essentially transforms the sample topography into a time domain signal. The bandwidth of this signal can be decreased by reducing the fast-axis scan frequency. Since the compensator ISR is called at a constant rate regardless of the scan frequency, additional “feedback points” will occur between the 256 recorded “image points” for slower scan frequencies. This is shown in Figure 5.5. In fact, the scan frequency is controlled by changing the desired number of feedback points per image point. Increasing the number of feedback points decreases the scan frequency, increases the real sampling rate in samples per unit length, and improves the performance of the system. There is a minimum of 256 feedback points per line, or one for each image point.

5.5 The modified controller software

The first changes made to the SPM controller software were intended to return the controller to its original functionality after the hardware upgrades, so that it could be used as an AFM or STM controller. A. Lemus assisted with these changes, which largely con-

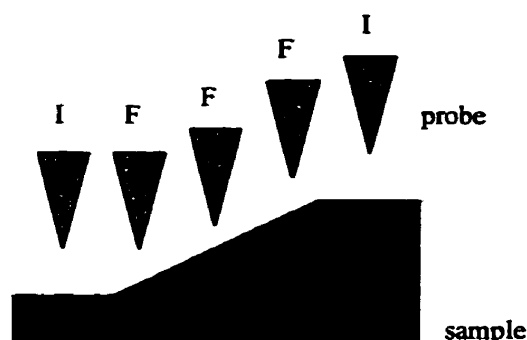


Figure 5.5 Feedback points (F) versus image points (I).

sisted of interfacing to the new ADCs and DACs. In addition, code was added to withdraw the probe whenever the scan size or location was changed, to prevent tip crashes. A subtle rounding error in the raster scanning algorithm that prevented lateral movement between some feedback points was also repaired. A significant problem discovered at this point involved the range of voltages applied to the piezoelectric scanner.

Piezoelectric scanners are manufactured inside an electric field. This field aligns the electric dipole moments of the individual crystals that make up the ceramic material, and makes the material expand or contract under an external electric field. However, a strong electric field applied in the right direction can cause the dipoles to randomize, or depole. This undesirable event greatly reduces the scanner's response to an applied electric field.

In the old controller, the high voltage amplifier for the z -electrode of the piezoelectric scanner used a different power supply from the amplifiers for the x - and y -electrodes. The voltage range of the z -electrode did not overlap the voltage ranges of the x - and y -electrodes, allowing the full ranges of the DACs to be used with no risk of depoling the scanner. However, the new controller hardware uses the same power supply and thus the

same voltage range (-200 V to $+200\text{ V}$) for all three axes. Since the manufacturer of the scanner commonly used in our laboratory specified that there was a risk of depoling the scanner if the z -electrode potential was more than 200 V above the x - or y -electrode potentials, some sort of voltage clamp was required.

The solution was to limit the z -electrode voltage based on the lowest potential applied to the x - and y -electrodes during a full image. This limit was calculated each time the image size or offset was changed. For image sizes less than the maximum, this gives a larger z -electrode voltage range than simply restricting the z -electrode potential to be below 0 V at all times. This dynamic voltage clamp is now used by the control program for all SPMs, except for one that has three separate piezoelectric actuators to manipulate the sample and does not need clamping. If necessary, this method could be extended further by recalculating the z -electrode voltage limit at each point based on the current x - and y -electrode potentials. It would also be possible to have different clamp levels for different piezoelectric scanners.

The next change made to the SPM controller software allowed two images to be acquired simultaneously. This was done by allocating the necessary additional memory on the DSP, using one of the surplus ADCs to acquire the second image, and transferring twice as much data to the PC for each raster line. This was again carried out in cooperation with A. Lemus. After making similar changes to the PC's user interface, the system was ready for use with the SRM.

The final change made to the control program was the most difficult. To maintain a constant flying height during an EFM scan, the sample topography must be known beforehand. The following algorithm was developed:

1. Perform a forward pass in contact AFM mode, to acquire the topographical data.
2. Return to the beginning of the line, still in contact AFM mode. Do not record any data during this pass.
3. Perform another forward pass, retracing the topography at a constant flying height to acquire the electrostatic force data.
4. Go to the next raster line, transfer the topographical and electrostatic force data for this line to the PC, and return to the beginning of the line in contact AFM mode.

Thus, the controller scans each raster line twice, first to obtain the topography and again to measure the electrostatic force. This technique, called lift mode, is illustrated in Figure 5.6.

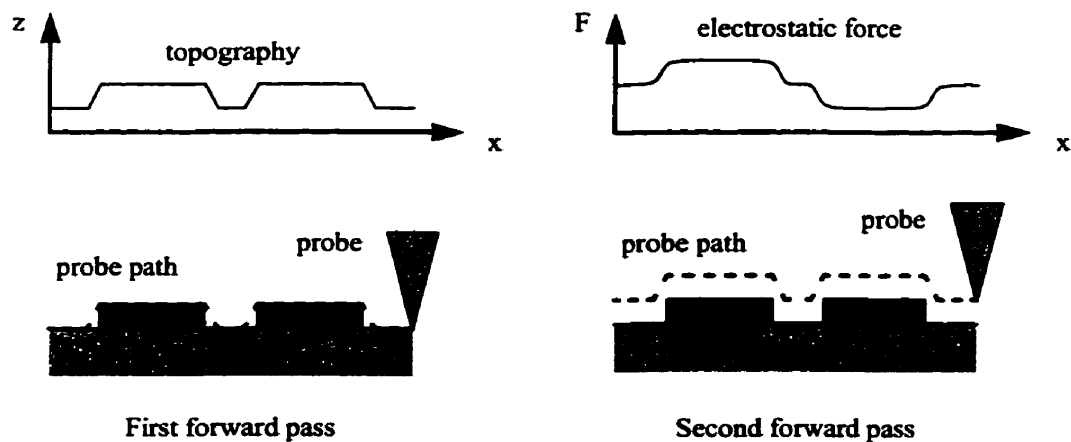


Figure 5.6 The lift mode technique.

The algorithm was integrated into the existing compensation ISR by adding a user-defined lift height parameter. If the lift height is zero, the ISR acts as an AFM/STM/SRM controller. If the lift height is greater than zero, the ISR acts as an EFM controller.

The implementation of the lift mode technique was complicated by the behaviour of the probe near the sample surface. Under ambient conditions, many surfaces have a very thin film of water on them. The capillary forces associated with this film are strong enough that they cause an SPM probe to adhere to the sample. When the probe and sample are pulled apart, they remain in contact until the spring force of the cantilever exceeds the capillary force and the probe abruptly “snaps off” the surface. This phenomenon is most noticeable in summer, when the relative humidity indoors is high.

The interaction of a contact mode AFM probe with a surface in air can be seen in a force versus distance curve. This curve is a plot of the probe-sample force as the separation between them is varied. In practice it is a plot of the cantilever deflection versus the z -position of the sample, so such curves are sometimes called $z(z)$ curves.

Figure 5.7 shows a typical force versus distance curve for a contact mode AFM as the scanner is ramped up and down. Initially, at point (a), the probe is far from the surface and there is no force between them. At point (b), the probe is very close to the surface and attractive van der Waals forces pull it downward into contact with the sample at point (c). From here, the surface deflects the cantilever upward until point (d). This would be a typical contact mode AFM operating point. Next, the scanner ramps back toward its starting position. The surface water layer holds the probe on the surface until the cantilever spring

force exceeds the capillary force and the probe snaps off the surface at point (e). After this point, there is no further interaction between the probe and the surface.

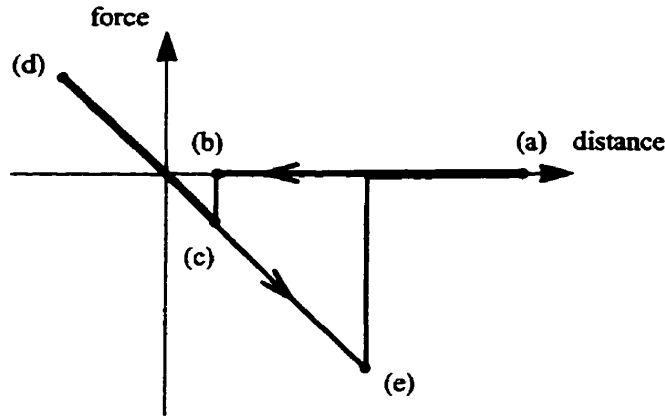


Figure 5.7 A typical force versus distance curve.

Figure 5.8 shows a force versus distance curve obtained during summer, when the relative humidity was approximately sixty percent. In this case the probe's initial and final positions were on the surface, where the cantilever deflection was appropriate for contact AFM. The piezoelectric scanner withdrew the probe from the surface as far as possible, then approached until the deflection sensor signal returned to its original value. The flat region around $1.3\ \mu\text{m}$ is simply clipping of the deflection sensor signal as it exceeds the range of the electronics used to measure it.

In this figure, the probe appears to return to the surface at a point closer than its starting position. This effect is due to hysteresis in the movement of the piezoelectric scanner. In addition, although it is not shown in the figure, after the scanner stopped the deflection sensor signal continued to increase because of scanner creep. Such nonlinearities in the scanner's transfer function cause errors in the calculated scanner position. Thus, the

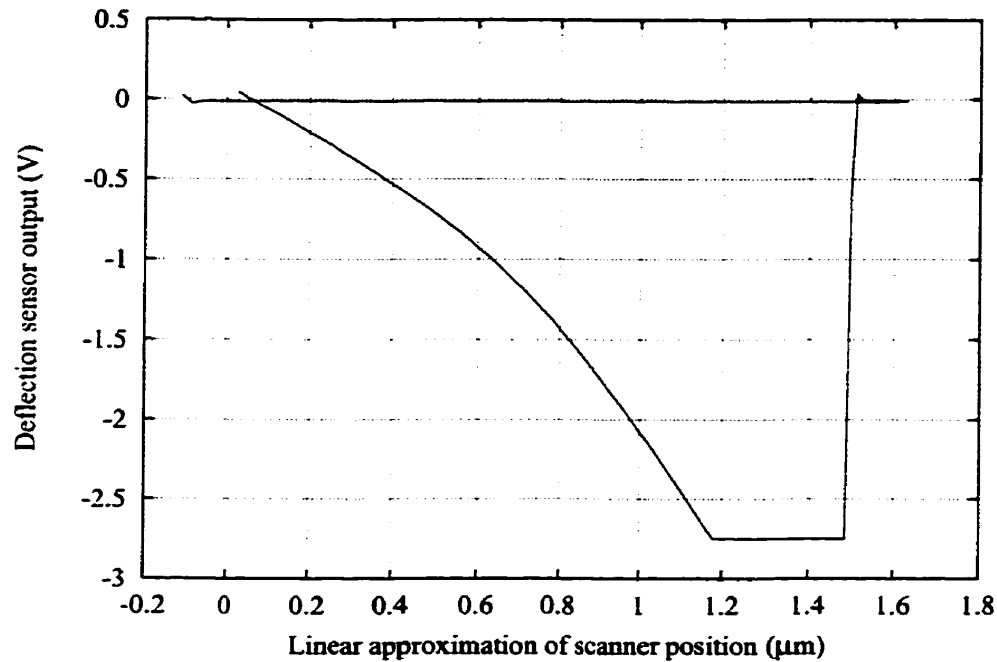


Figure 5.8 An example of a force versus distance curve.

horizontal axis of the plot is only a linear approximation of the scanner position based on the applied voltage.

The behaviour of SPM probes near a surface complicates the question of how to position a probe at a known flying height above a sample. The probe cannot simply be moved directly to its final height, since this height may not be enough to pull the probe off the surface. Also, the scanner potential that corresponds to the desired flying height depends on the scanner potential at which the probe touches the surface with zero cantilever deflection.

The complete lift mode algorithm addresses these problems and is described in detail below.

1. Perform a forward pass in contact AFM mode, to acquire the topographical data.

2. Return to the beginning of the line, still in contact AFM mode. Do not acquire any data during this pass.
3. Acquire a force versus distance curve while completely withdrawing the probe and returning it to its original deflection. The scanner performs up to 256 steps of 1.563 V in each direction, approximately 200 μ s apart. The deflection sensor is sampled at each step, as well as before and after the scanner is ramped.
4. Find the cantilever deflection sensor signal level that corresponds to no probe-surface contact and thus zero deflection, by backtracking the approach part of the force versus distance data from point (c) in Figure 5.9 until two consecutive pairs of points are less than 1 mV apart, at point (a). This gives a reasonable noise margin.

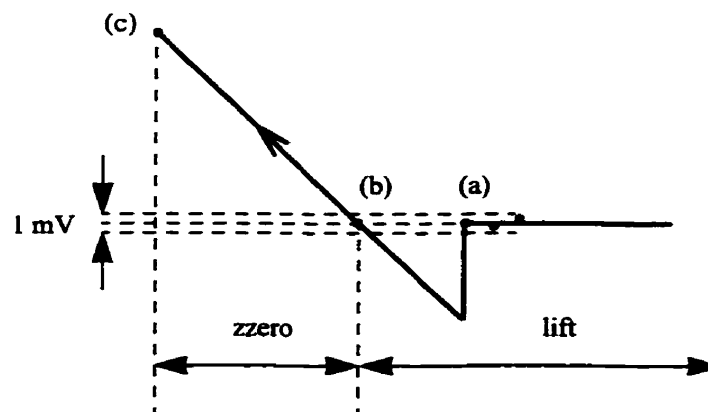


Figure 5.9 Finding the offset between the topographical data and the sample surface.

5. Find the scanner z-electrode potential that corresponds to zero lift height. Again, backtrack the approach part of the force versus distance

data from point (c) in Figure 5.9 until the cantilever deflection signal is less than or equal to the value found in the previous step, at point (b).

The scanner potential difference between this point and the point where the cantilever deflection returns to its AFM operating level is the scanner z -electrode potential offset between the topographical data and the sample surface. This offset is labelled “ z_{zero} ” in Figure 5.9.

6. Completely withdraw the probe again to release it from the capillary forces, and return it to a height above the surface specified by the user. This scanner potential is the potential at the AFM operating point, plus “ z_{zero} ,” plus a linear approximation of the scanner potential difference that corresponds to the desired lift height.
7. Retrace the recorded forward pass at this constant height above the surface to acquire the electrostatic force data.
8. Go to the next raster line, transfer the data for this line to the PC, and return to the beginning of the line in contact AFM mode.

Finding the scanner z -electrode potential offset between the topographical data and the sample surface as frequently as every pass reduces the effect of drifts in the cantilever deflection sensor system. These drifts can shift the $z(z)$ curve vertically while the controller maintains the same cantilever deflection in AFM mode, changing the offset.

When acquiring the $z(z)$ curve, the controller must know whether an increasing or decreasing z -electrode potential withdraws the probe. Fortunately, a decreasing z -electrode potential withdraws the probe on all of the microscopes in the laboratory, so the controller

assumes this scanner polarity. This polarity also eliminates any added risk of depoling the scanner in lift mode since the z-electrode potential never exceeds its value in the first forward pass, except for the effects of hysteresis when ramping back to the AFM operating point.

When finding “zzero,” the controller expects an increasing cantilever deflection sensor signal to correspond to an increasing cantilever deflection. This relationship is easily adjusted with some simple changes to the sensor’s electronics.

During the second forward pass, only the image points of the topographical data are available for retracing. The scanner potentials at the feedback points are calculated by linear interpolation. Also, if the desired lift height pushes some points outside the scanner’s range, those points are simply clipped before being sent to the DAC, and no warning is given to the user.

Finally, one minor change was made for the EFM. The lock-in amplifier used with the EFM often overloaded when the probe was lifted off the surface, and did not settle until well into the raster line. This problem was solved by feeding data from one of the controller’s ADCs directly to one of the unused DACs while acquiring electrostatic force data, essentially making a switch synchronized with the compensator. A delay was also added at the start of the second scan to allow the lock-in amplifier to settle. This is only a temporary solution, and is expected to be replaced by a hardware switch.

Figure 5.10 shows a detailed flowchart for the completed compensator ISR. Source code for all of the routines is given in Appendix A.

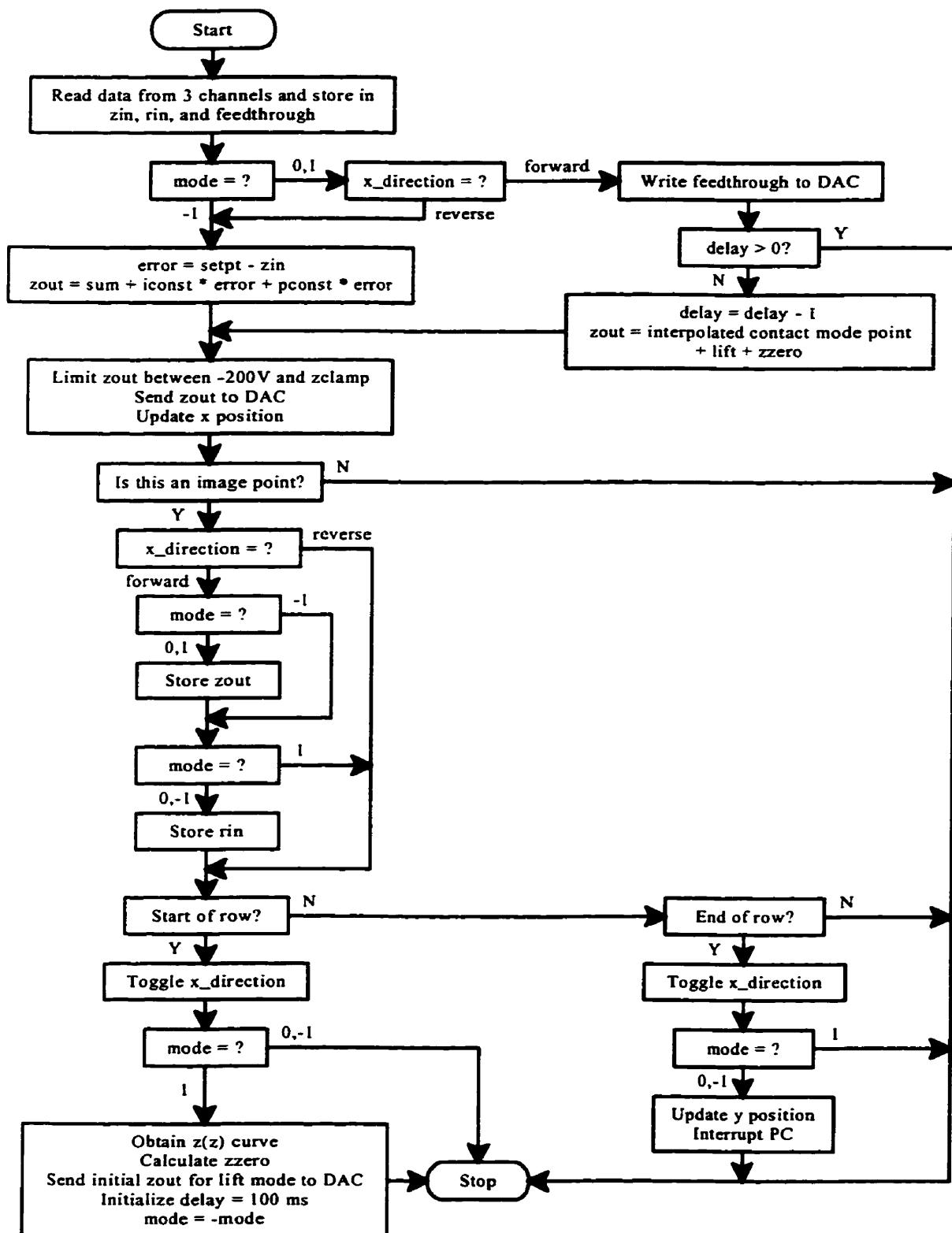


Figure 5.10 Flowchart for the modified compensator routine.

A special version of the DSP software was written for the microscope with three independent piezoelectric actuators mentioned earlier. The only differences in this version are its lack of a dynamic voltage clamp, and a ramp change in lateral position instead of a step change when the scan size or offset is changed. This ramp was part of a failed attempt to linearize the lateral motion of the piezoelectric scanner by operating the x - and y -axis actuators with feedback control.

5.6 Results and discussion

As previously mentioned, the performance of the SPM control system can be improved by decreasing the scan frequency. However, a low scan frequency can make the time needed to scan an entire image unacceptably long. A better way to improve the system's performance is to increase the rate at which the compensator ISR is called.

The rate at which the compensator ISR is called is limited by its execution time. The ISR's execution time in various modes was checked using Motorola's DSP56000 simulator software (SIM56000), and the results are shown in Table 5.1. These times will vary slightly if the signal sent to the piezoelectric scanner must be clamped, or if the scanner is at the beginning or end of a scan line. The execution time at the start of the lift mode second pass is especially long, since acquiring the $z(z)$ curve can take up to 200 ms.

The DSP is programmed to interrupt itself every 640 clock cycles. Although this is less than the longest execution time in Table 5.1, this does not cause a problem. In fact, it is possible to decrease the interrupt rate to just above the largest average of the ISR execution times at feedback points and image points $((480 + 650) / 2 = 565$ clock cycles in

Table 5.1 Compensator routine execution time during forward passes.

Conditions	Clock cycles (Ccyc)
Contact mode, feedback point	444
Contact mode, image point	640
Lift mode first pass, feedback point	444
Lift mode first pass, image point	614
Lift mode second pass, feedback point	480
Lift mode second pass, image point	650

this case). Additional interrupts at the same priority are disabled during interrupt service, so the interrupt request that would occur near the end of the ISR at an image point simply becomes pending. When the routine finally terminates, if no other interrupts at the same priority are pending the ISR is immediately called again. Each image point is followed by at least one feedback point, so this must be a feedback point. Since the sum of the feedback point and image point ISR execution times is less than two timer periods, the DSP can make up for the time lost servicing the image point interrupt before the next interrupt request. Since both the lateral and vertical position of the scanner are controlled by the compensator ISR, the number of samples per unit length during a scan line remains constant. The scanner's motion may be distorted in the time domain, but this is probably unimportant. The interrupt rate can be adjusted in increments of 64 clock cycles.

One factor that affects the ISR's execution time is the number of wait states inserted during memory accesses. The application development board containing the DSP has high-speed memory and requires no wait states, but the controller's ADCs, DACs, and digitally programmable filter are slower. They are mapped into the same memory space as the DSP program's variables, which forces all program variables and external devices to be accessed with the same number of wait states. To improve the ISR's performance the

external devices should be remapped to the memory space reserved for external devices, which has its wait states set separately. The program variables could then be accessed with no wait states while the external devices retained additional wait states. The DSP is currently configured to add four wait states when accessing program variables. An estimate using SIM56000 shows that this change would reduce the ISR's execution time by several microseconds. Remapping the external devices requires alterations to the FPGA's address decoder and was beyond the scope of this project.

It is difficult to accurately measure the flying heights produced by the lift mode algorithm. However, some sources of error can be described. For example, suppose the controller cannot find a flat region that corresponds to zero deflection in the force versus distance data because of long-range forces or other effects. It will then continue backtracking the data until it finds the deflection sensor samples taken before the scanner was ramped, which will almost certainly be constant. It will calculate "zzero" to be about zero, and the flying height will be smaller than expected. Scanner creep after the final approach ramp will probably also cause the flying height to be less than intended.

The DSP software assumes that an increasing z-electrode potential corresponds to probe-sample approach. This assumption is more difficult to meet through additional electronics than the deflection sensor polarity assumption, because of the high voltages involved. Given that most of the SPMs in the laboratory were designed in-house, it may be easiest to simply design any future SPMs with this in mind. If an SPM with the opposite polarity must be used, the DSP program would need significant changes to allow a choice. Creating a separate version of the program for this polarity is an alternative.

The dynamic clamp on the z -electrode potential may be too conservative, especially for large scan sizes where it is quite restrictive. It may be useful to give the user the option of disabling the clamp, or changing the clamp level. This feature would also make it practical to combine the clampless version of the software with the generic version.

Both the dynamic clamp and the lift mode algorithm assume that a large positive z -electrode potential can depole the scanner. Again, giving the user a choice of polarity would be awkward.

The switch added for the EFM is probably unnecessary. The delay at the start of the lift mode second pass is long enough to let the lock-in amplifier to settle without using the switch, if the lock-in amplifier is set up so that it does not overload too badly.

The revision of the DSP software was a substantial task. It required learning about microprocessor interfacing, the syntax of the DSP's assembly language, and the eccentricities of the DSP and its support software. For example, the DSP uses fractional arithmetic. It assumes that the values stored in its registers are two's complement signed fractions, with an implicit decimal point following the sign bit. In this system, squaring a 4-bit signed binary number such as 0010 does not produce 00000100 as expected, but rather 00001000. Because of the effort put into changing to the DSP software, it makes up a larger part of this thesis than anticipated.

The members of the SPM group have extensively tested the modified SPM controller. The remainder of this section briefly showcases a few images they have obtained.

Scanning resistance microscopy maps surface resistivity with high resolution, and can provide information about semiconductor dopant distributions for process modelling

and diagnostics. The simultaneously acquired AFM and SRM images in Figure 5.11 were obtained with the contact mode of the modified controller.

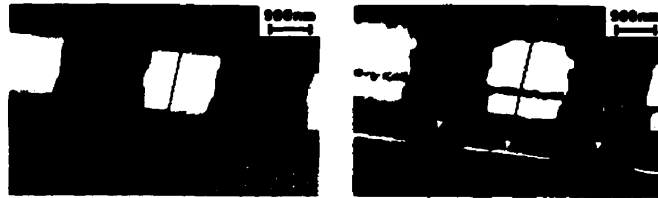


Figure 5.11 Simultaneously acquired AFM (left) and SRM images of a MOSFET cross-section [26].

Magnetic force microscopy uses a magnetized probe to measure magnetic field variations across a surface. The information helps manufacturers of magnetic data storage devices improve the speed and storage density of their products. The concurrently acquired AFM and MFM images in Figure 5.12 show polish marks and magnetic bits on a hard disk platter. These images were acquired with a fast-axis scan frequency of 1.5 Hz and a flying height of 100 nm in lift mode.

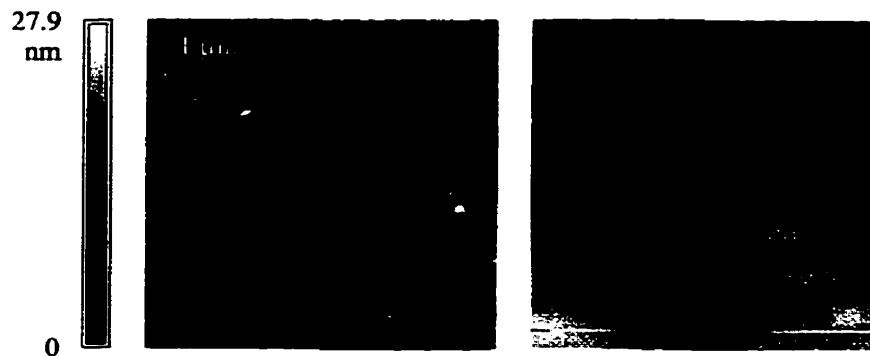


Figure 5.12 Concurrently acquired AFM (left) and MFM images of a hard disk platter.

The sample images in this section show that the modified DSP software is an effective controller for the various SPMs used in the department's laboratory. Some EFM images are also given later in this thesis.

6

Changes to the user interface

6.1 Introduction

The DSP software is closely linked to a PC-based user interface. This software must also be modified before the digital SPM control system can control an EFM or SRM. The revised user interface must be able to receive, display and save two sets of image data from the DSP. It must be able to supervise various SPMs that may behave somewhat differently. In future, the user interface may also need to be extended to acquire three images. This possibility was not discussed in the previous chapter because the control of an SPM that generates three simultaneous images was beyond the scope of this project.

This chapter describes the existing user interface and how it was changed to match the revised DSP software. The results are briefly discussed, and some possibilities for future development are explored.

6.2 The existing user interface

An essential part of the University of Manitoba's digital SPM control system is its graphical user interface (GUI). This program was written in Borland Turbo Pascal by K. Yackoboski and G. McGonigal, and runs under Microsoft DOS on an IBM-compatible PC. It communicates with the DSP software through a customized prototyping board attached to the PC's Industry Standard Architecture (ISA) bus and one of the DSP's ports. The program transfers the user's commands to the DSP, and stores acquired data on the PC's hard drive. The stored data are typically analyzed later on a workstation.

The user interface has two states: review mode and acquire mode.

In review mode, the PI compensator routine on the DSP is disabled. Images can be loaded from or saved to disk, and control system parameters can be changed by typing in new values on the keyboard. This mode is used to setup the SPM before scanning and to review previously acquired images. A screen capture of this mode is given in Figure 6.1.

In acquire mode, the PI compensator routine is enabled and the user interface regularly receives data from the DSP. Some of the features available in review mode are disabled, and control system parameters can only be changed incrementally using the mouse. This mode is used for data acquisition. A screen capture is shown in Figure 6.2.

The user interface consists mainly of a set of buttons. By clicking a few buttons with the mouse, the user can change the size and location of the area being scanned, the proportional and integral gains of the PI compensator, the fast-axis scan frequency, the probe bias voltage for STM, and the desired cantilever deflection or probe current (com-



Figure 6.1 The digital SPM controller's graphical user interface, in review mode.

monly called the set point). The probe can also be landed automatically by running the coarse positioning stepper motor found on many SPMs until the deflection signal or probe current exceeds the set point, if the sensor signal increases with increasing cantilever deflection or probe current.

The user interface sends commands to the DSP software by interrupting the DSP. The corresponding interrupt service routine on the DSP initiates the transfer of any data needed to carry out the command.

Part of the user interface program itself is an interrupt service routine. When the DSP has acquired a complete raster line of 256 data points, it interrupts the PC. This routine then transfers the data from the DSP and stores it in the PC's memory. The 16-bit data

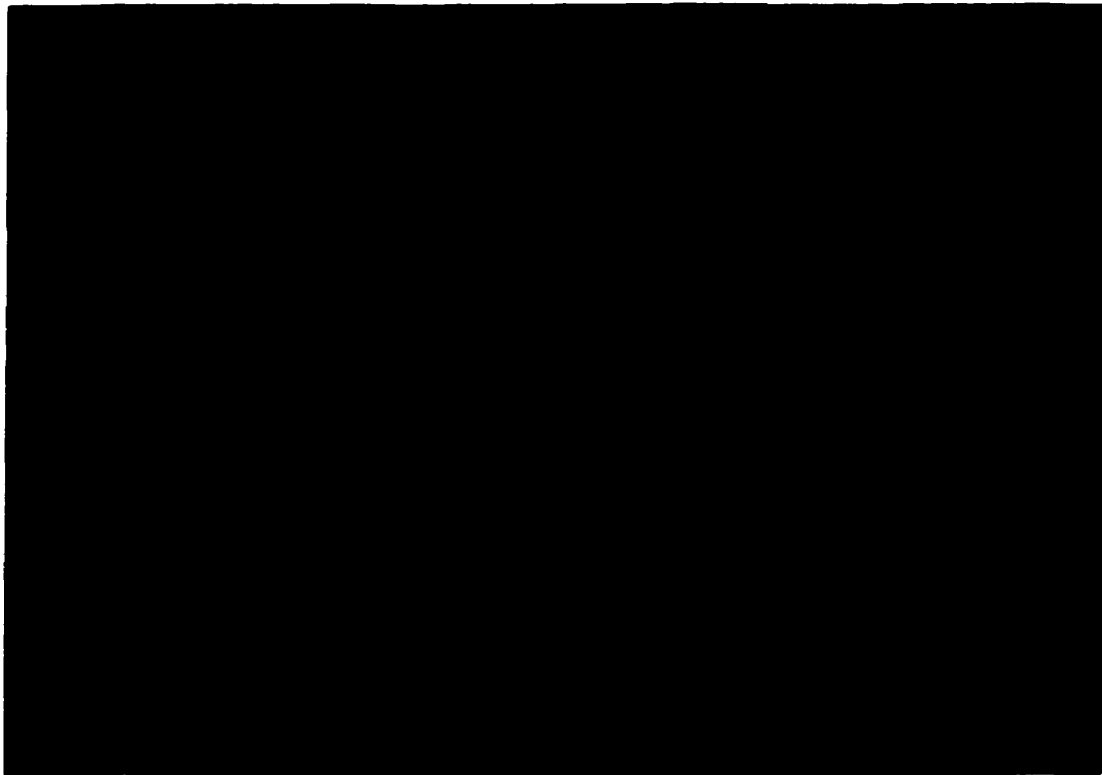


Figure 6.2 The digital SPM controller's graphical user interface, in acquire mode.

are then mapped to the 64 greyscales available on the PC and displayed on the screen as part of an image. Each line is also displayed as an oscilloscope trace. While the image displayed on the screen is continuously overwritten by new data, the last 512 lines acquired are always stored in the PC's memory. This "swinging buffer" allows the last complete 256-line image acquired to be saved to the PC's hard disk at any time. A saved image consists of an array of 65 536 16-bit integers, and a separate file containing information about the image such as its size, bias voltage, and a brief description.

Before displaying each line in acquire mode, the slope and offset of the line can optionally be removed. This option is usually enabled because the slope and offset of an AFM or STM image depend on the relative positions of the probe and sample, and are not

meaningful. Also, each line is independently scaled to the full range of 64 greyscales before display regardless of its actual range. Although the resulting display is a distortion of the actual data, it usually enhances the features of the image. The data stored to the PC's hard disk are not processed in any way.

6.3 The revised user interface

The proposed changes to the user interface seem fairly straightforward. However, some subtle difficulties were encountered.

For compatibility reasons, DOS uses the real mode of the PC's microprocessor. In this mode, the microprocessor can directly access a maximum of 1 MB of memory, regardless of the amount of memory actually present in the PC. Some 384 KB of this is reserved, leaving 640 KB of so-called conventional memory for the code, data, and run-time stacks of all resident programs, including DOS. The user interface uses overlays to effectively increase the available memory. Overlays allow a program to swap subroutines that are not currently being used to nonconventional memory or disk.

The user interface maintains the last 512 raster lines acquired in memory. This buffer requires 256 KB per image. Storing two such image buffers in memory would require 512 KB, leaving 128 KB for the user interface's code, stack, other data, and all other programs. This was believed to be insufficient, especially if three images may be acquired in future.

The solution was to recompile the program using Borland Pascal with Objects 7.0 [27]. This compiler can create programs that access the protected mode of

Intel 80286 and newer microprocessors, while retaining access to DOS services. In protected mode, applications can address up to 16 MB of memory. Other options were considered, such as using nonconventional memory to store data, but this method needed almost no reprogramming. The protected mode interface replaced the overlay scheme.

Although the new compiler solved the memory problem, it also introduced a new one. The third party high resolution graphics driver that interfaces the Borland graphics library routines with the PC's display hardware was not compatible with protected mode. This problem was easily solved by downloading a different graphics driver [28] from Borland's FTP site. The new driver uses the VESA BIOS Extension (VBE) standard to access high resolution graphics modes, so the modified user interface should work on any PC that complies with this specification.

The increased dynamic range of the new SPM controller hardware caused another problem. The old controller hardware had a 16-bit DAC driving the vertical axis of the SPM's piezoelectric scanner. Thus, a constant force AFM or constant current STM image consisted of an array of 16-bit integers. The new controller hardware has 20-bit DACs and 18-bit ADCs. As a result, when operating an SRM or EFM in constant force mode the topographical data are 20 bits wide and the resistance or electrostatic force data are 18 bits wide. Images acquired by the new hardware can no longer be stored as arrays of 16-bit integers without loss of precision. This limitation is not a problem for the DSP, which uses 24-bit words, but it is a problem for the PC, which uses 16-bit words. Although 32-bit "dwords" could be used on the PC for calculations and storage, the resulting images would

need twice as much memory in RAM and on disk. The laboratory's data analysis software would also need changes.

The following compromise was reached. Since RAM is less scarce with the new compiler, a 512 KB buffer was allocated for each image. The 20-bit topographical data are stored as arrays of 32-bit integers. However, the costs of saving images to disk in this format outweighed the benefits, especially considering that data analysis largely consists of simply looking at an image. The four bits of increased resolution did not justify doubling the storage space for each image and modifying the data analysis software. Therefore, before topographical images are saved to disk they are reduced to sixteen bits by the following process.

First, a plane is fitted to the topographical data and then subtracted from it. Any points that are pushed outside the 20-bit range are truncated. This "plane subtraction" removes any overall slope from the data and centres it about zero. Since the slope and offset of the topographical data depend on the orientation of the probe with respect to the sample, little information is lost during this process. The data are then shifted bitwise to the right until all of the data points fall within a 16-bit range. This method is better than simply shifting four bits to the right, since the data may need less than twenty bits for storage after plane subtraction. The sixteen most significant bits containing information are retained, minimizing the information lost in reducing the image to sixteen bits. The scale of the image is adjusted accordingly in the file containing the image description.

The resistance or electrostatic force image acquired by the controller is also reduced to sixteen bits, although not by the same process since the slope and offset of this

image are not arbitrary. In this case, the data are simply shifted two bits to the right immediately after transfer from the DSP, losing two bits of precision. Since the input range of the ADC is 5.5 V, this effectively reduces the ADC resolution from 21.0 μV to 83.9 μV , which is probably below the noise floor of the attached equipment anyway. The buffer for this image is the same size as the topographical image buffer so that the code for saving, loading and displaying images can be reused.

The remainder of the changes made to the user interface were either usability improvements or minor bugfixes. The layout of the screen was rearranged to accommodate a second image and virtual oscilloscope. The dynamic range of the topography virtual oscilloscope was increased, with the scales set to the 1:2:5:10 ratio common on real oscilloscopes. The mouse pointer and buttons were changed to resemble the familiar Microsoft Windows mouse pointer and buttons. Arrows were added to buttons that permit incremental changes in acquire mode. The incremental changes made by some of these buttons were changed to percentages of the current value instead of a small constant value. Controls for loading and saving a second image, changing the current directory, setting the lift height, changing the digitally programmable filter cutoff frequency, disabling the display of one or both images, and acquiring a force versus distance curve were added. An alphabetized, scrolling directory listing was added to the load and save dialogs. Image saving is now disabled when space is very low on the current disk. A bug that caused the first row of each image to be saved incorrectly was fixed. The use of the right mouse button was largely discontinued, and the stepper motor controls were clarified.

The dynamic ranges of the PI compensator proportional and integral gain settings were increased. They were also allowed to be both positive and negative, although the SPMs used in our laboratory normally require negative gains. The minimum fast-axis scan frequency was lowered to 1 mHz.

The “mouse dropping” problems [25] that have always plagued the user interface were partially fixed. Short lines that appeared below the mouse were removed by correcting the size of the array holding the mouse image. Tests showed that the second problem of partial mouse images being left behind when the mouse image is overwritten onscreen by a row of data or a virtual oscilloscope trace could be fixed by hiding the mouse when writing to the screen, but this caused excessive flickering of the mouse. This problem is not serious, so it was ignored.

The DSP timer period, piezoelectric scanner parameters and stepper motor polarity are now read from a configuration file specified when the user interface is started, instead of being coded directly into the program. This change allows the SPM hardware to be altered without recompiling the user interface. Batch files were created to program the FPGA and DSP, and run the user interface with the appropriate parameters for several different microscopes.

The ability to save 8-bit greyscale uncompressed TIFF images [29] was added to simplify publishing. The image description is stored as a comment in the TIFF file, although few image viewers can access this information. One software that can is XV [30].

The workstation-based data analysis software and image file format used in the laboratory are legacies of an earlier analog STM control system, and were designed to be used only with that STM. As a result, the files express the image dimensions in terms of the piezoelectric scanner voltage ranges and a z-axis gain factor. For each SPM, a simple program converts the image data into a form the data analysis software can understand, and scales the image dimensions appropriately. The data analysis software then uses a fixed ratio to calculate the physical dimensions of the image. The revised control system will be used with many different SPMs, so the user interface was modified to store the physical dimensions of the image directly, along with the lift height. This change allows a single new program to translate the image data and dimensions from any SPM. It also makes it easier to read the dimensions manually, if an image needs to be viewed with some other software. The updated image file format is described in Appendix B.

6.4 Results and discussion

Figures 6.3 and 6.4 show screen captures of the final version of the digital SPM controller graphical user interface.

On a 486DX2/66, the modified user interface is not fast enough to display both images in acquire mode at fast-axis scan frequencies above 6.94 Hz. No data is lost at these speeds, but some scan lines are not drawn. Disabling one of the images allows the other to be fully displayed in acquire mode up to the system's maximum scan frequency of 31.25 Hz. This effect is probably caused by the graphics system. For example, an examination of the source code for the third party graphics driver indicates that screen coordi-

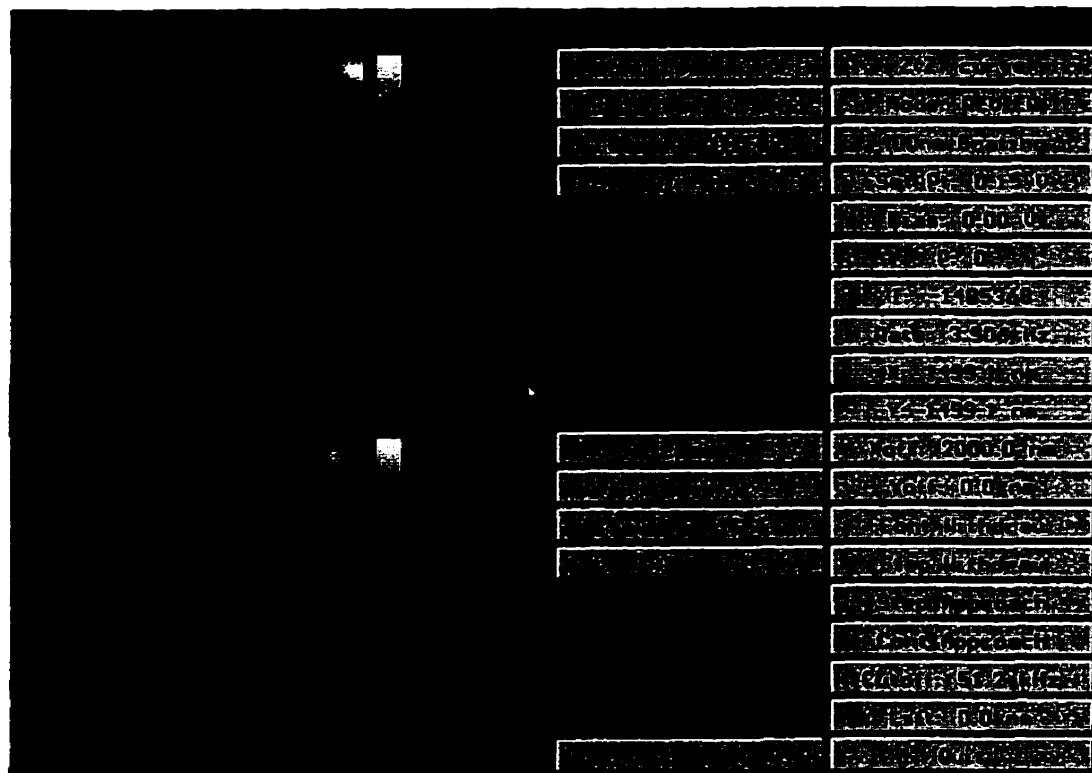


Figure 6.3 The new digital SPM controller graphical user interface, in review mode.

nates are mapped into memory by simply multiplying the y -coordinate by the number of pixels per row and adding the x -coordinate. Multiplication is time-consuming. A faster way would be to use a lookup table containing the products of all possible y -coordinates and the number of pixels per row. Another possibility is that the protected mode interface switches the processor to real mode every time it writes to the screen, which would also be time-consuming. Fixing either of these issues requires a more detailed understanding of DOS programming.

As with the DSP software, the changes to the digital SPM controller user interface made up a larger part of this thesis than anticipated. Extending the swinging buffer to handle two images, and adjusting the various controls and displays to work with more than

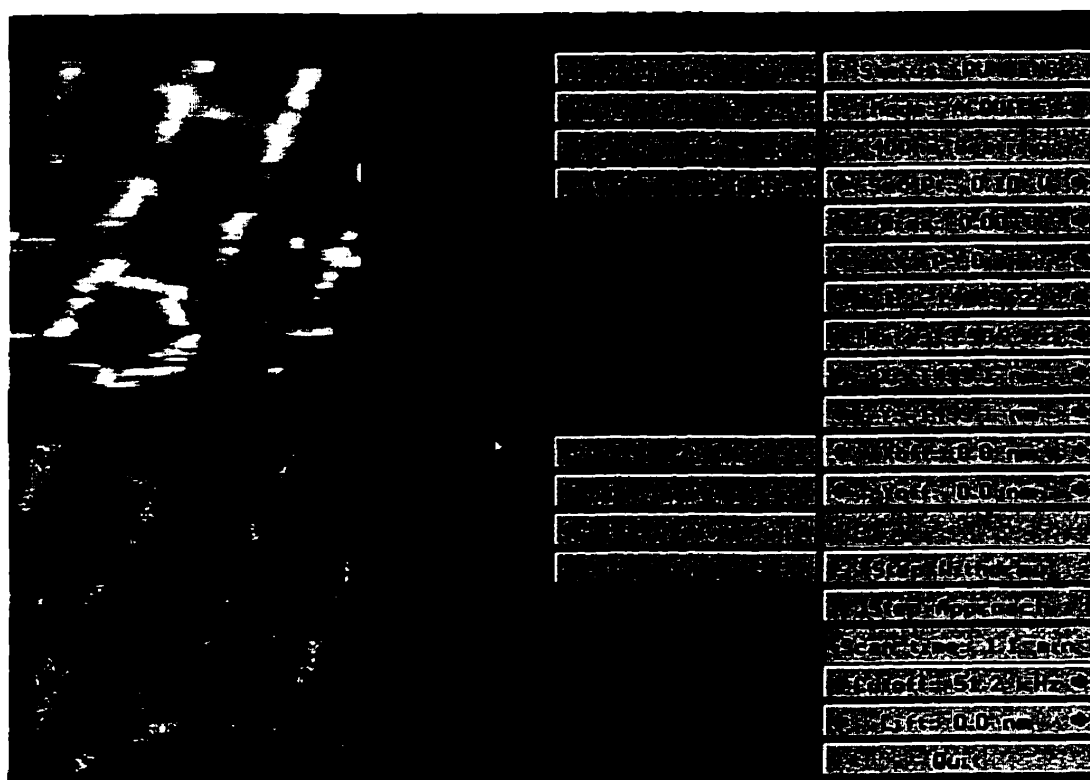


Figure 6.4 The new digital SPM controller graphical user interface, in acquire mode.

one piezoelectric scanner were challenging problems. There were also some hardware-related issues to be dealt with such as the conversion between the PC's 32-bit long integers and the 20-bit two's complement codes used by the new DACs, versus the 16-bit offset binary codes used by the old DACs.

6.5 Future development

As mentioned in the previous chapter, a user-defined clamp level may be a useful addition to the DSP software. The user interface would then need a corresponding change, either in the form of an additional parameter in the configuration file or another button.

In future, the digital SPM control system may be extended to simultaneously acquire three images. Using Borland Pascal with Objects 7.0 to generate a protected mode

application, there is certainly enough memory available on a typical PC for a third image buffer in the user interface. However, the user interface's present layout is too crowded to display three images on the screen simultaneously.

There are four possible solutions. Switching the display to a higher resolution could free enough space on the screen, but would probably require a graphics controller with more memory and an expensive, large-screen monitor for acceptable image quality and size. An alternative is to replace the button-based interface with a menu-based interface. This would probably be tedious to implement. On the other hand, the user interface could be ported to a different operating system. Under an operating system such as Microsoft Windows, it may be possible to display each image in a separate window. This would involve extensive changes to the code because of the different application programming interface. It would also require a new compiler, since Borland Pascal with Objects 7.0 cannot be used to write interrupt service routines for Microsoft Windows [31]. It might even require rewriting the program in a more popular language such as C, since Pascal compilers are now rare. Finally, it may be possible to add a second monitor to the system and separate the image display from the controls. Some commercial SPM control systems use this technique.

The laboratory's custom workstation-based data analysis software uses a windowing system that is no longer supported by the manufacturer. The department's network administrators want to upgrade the few remaining workstations that use the old system. As a result, this software will likely be upgraded or replaced in the near future.

If the software is upgraded, its inherent bias toward the STM it was designed to support should be removed. This change would eliminate the need for a translation program between the user interface and the data analysis program. The image file format should also be revised. Currently, the image file format used in the laboratory is composed of a binary data file and a separate ASCII parameter file. This two-file system complicates data transfer between platforms, and a single-file format containing both the data and the parameters is preferable.

If the software is replaced by a commercial program, an industry standard file format should be adopted. The popular graphic file formats such as TIFF are inadequate since they typically don't support 16-bit greyscale images, but a scientific data format such as the National Center for Supercomputing Applications' hierarchical data format (HDF) [32] may be acceptable. One of the commercial SPM manufacturers' file formats would also be acceptable, if the specification is available.

7

Phase modulated electrostatic force microscopy

7.1 Introduction

The usual method of measuring signals inside integrated circuits with the electrostatic force microscope at the University of Manitoba is to amplitude modulate a probe signal that is synchronized with the circuit signal. However, other types of modulation are possible. This chapter examines the effects of modulating the phase of the probe signal, and describes some measurements performed with this technique. The results are discussed, and some future experiments are suggested.

7.2 Theory

Finding a general expression for the deflection of an EFM probe carrying a phase modulated signal is difficult. However, it is possible to find a simple expression in specific cases.

Suppose the modulating signal is a 50% duty cycle square wave. The probe signal can then be expressed as

$$v_p(t) = s(t)v_s(t) + (1 - s(t))v_s(t - \tau) \quad (7.1)$$

where $s(t) = \frac{1}{2} + \frac{2}{\pi} \sum_{n=1}^{\infty} \frac{1}{n} \sin \frac{n\pi}{2} \cos n\omega_r t$ is a 50% duty cycle square wave with extrema of

0 and 1 and a fundamental frequency equal to the cantilever resonant frequency ω_r . The sampling signals $v_s(t)$ and $v_s(t - \tau)$ are identical except for the phase shift τ and have fundamental frequency $\omega_0 \gg \omega_r$. Essentially, the two sampling signals are alternately enabled and disabled by the two modulating signals. Note that by fixing the probe signal modulation, we are disallowing the force nulling measurement scheme.

Substituting into (4.4), the electrostatic force on the cantilever is

$$F_z(x, y, z, t) = \frac{1}{2} \frac{\partial}{\partial z} C(x, y, z) [s(t)v_s(t) - s(t)v_s(t - \tau) + v_s(t - \tau) - v_c(x, y, t) + \phi]^2. \quad (7.2)$$

Expanding,

$$\begin{aligned} F_z(x, y, z, t) = & \frac{1}{2} \frac{\partial}{\partial z} C(x, y, z) [s^2(t)v_s^2(t) + s^2(t)v_s^2(t - \tau) + v_s^2(t - \tau) + v_c^2(x, y, t) \\ & + \phi^2 - 2s^2(t)v_s(t)v_s(t - \tau) + 2s(t)v_s(t)v_s(t - \tau) - 2s(t)v_s(t)v_c(x, y, t) + 2s(t)v_s(t)\phi \\ & - 2s(t)v_s^2(t - \tau) + 2s(t)v_s(t - \tau)v_c(x, y, t) - 2s(t)v_s(t - \tau)\phi - 2v_s(t - \tau)v_c(x, y, t) \\ & + 2v_s(t - \tau)\phi - 2v_c(x, y, t)\phi] \end{aligned} \quad (7.3)$$

Realizing that $s^2(t) = s(t)$, this reduces to

$$\begin{aligned}
F_z(x, y, z, t) = & \frac{1}{2} \frac{\partial}{\partial z} C(x, y, z) [s(t)v_s^2(t) - s(t)v_s^2(t-\tau) + v_s^2(t-\tau) + v_c^2(x, y, t) + \phi^2] \\
& - 2s(t)v_s(t)v_c(x, y, t) + 2s(t)v_s(t)\phi + 2s(t)v_s(t-\tau)v_c(x, y, t) - 2s(t)v_s(t-\tau)\phi \\
& - 2v_s(t-\tau)v_c(x, y, t) + 2v_s(t-\tau)\phi - 2v_c(x, y, t)\phi]
\end{aligned} \quad (7.4)$$

Replacing $s(t)$ with its Fourier series and dropping all components except those at ω_r , this becomes

$$\begin{aligned}
F_z(x, y, z, t)|_{\omega = \omega_r} = & \frac{1}{\pi T} \frac{\partial}{\partial z} C(x, y, z) \\
& \{ \int_0^T [v_s^2(t) - v_s^2(t-\tau) - 2(v_s(t) - v_s(t-\tau))v_c(x, y, t) + 2(v_s(t) - v_s(t-\tau))\phi] dt \} \\
& \cos \omega_r t
\end{aligned} \quad (7.5)$$

where $T = \frac{2\pi}{\omega_0}$ and $\frac{1}{T} \int_0^T f(t) dt$ is the mean of $f(t)$ over its period T .

Now, since $v_s^2(t)$ and $v_s^2(t-\tau)$ are identical except for phase,

$\int_0^T [v_s^2(t) - v_s^2(t-\tau)] dt$ vanishes. Similarly, $\int_0^T 2[v_s(t) - v_s(t-\tau)]\phi dt$ vanishes. The resulting cantilever deflection at ω_r is

$$\Delta z(x, y, z, t)|_{\omega = \omega_r} \approx -\frac{2Q}{\pi k T} \frac{\partial}{\partial z} C(x, y, z) \{ \int_0^T [v_s(t) - v_s(t-\tau)] v_c(x, y, t) dt \} \cos \omega_r t. \quad (7.6)$$

A comparison to (4.8) shows that this is identical to the cantilever deflection that would result from amplitude modulating a sampling signal $v_s(t) - v_s(t-\tau)$ with $A = 0$

and $K = -\frac{2}{\pi}$. A 50% duty cycle square wave modulating signal with extrema of $\pm\frac{1}{2}$ and phase opposite to $s(t)$ has these parameters.

7.3 Rectangular pulse sampling

Suppose $v_s(t)$ is a train of rectangular pulses of width δ and height V_s ,

$$v_s(t) = \sum_{n=-\infty}^{\infty} V_s G_{\delta}(t - nT - t_0) \quad (7.7)$$

where $T = \frac{2\pi}{\omega_0}$ and t_0 is a controllable delay. As shown in Figure 7.1, for $\tau < \delta$ the equivalent amplitude modulated sampling signal is a train of rectangular pulses of width τ and height V_s which alternate in polarity.

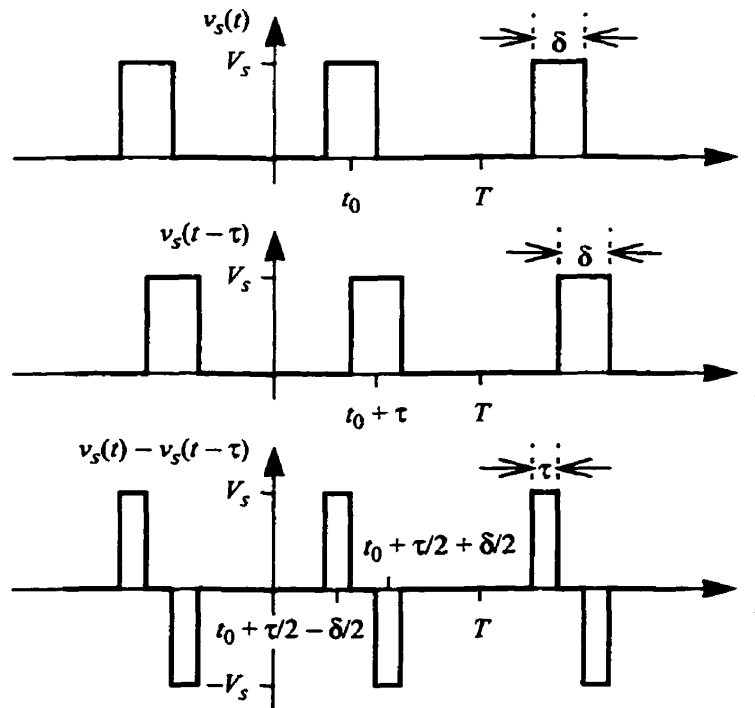


Figure 7.1 Equivalent phase modulated (top, middle) and amplitude modulated sampling signals.

Substituting into (7.6) and realizing that

$$\int_0^T [v_s(t) - v_s(t - \tau)] v_c(x, y, t) dt \approx V_s \tau \left[v_c(x, y, t_0 + \frac{\tau}{2} - \frac{\delta}{2}) - v_c(x, y, t_0 + \frac{\tau}{2} + \frac{\delta}{2}) \right] \quad \text{for}$$

$\tau \ll T$, the cantilever deflection becomes

$$\begin{aligned} \Delta z(x, y, z, t) |_{\omega = \omega_r} \\ \approx -\frac{2QV_s\tau}{\pi kT} \frac{\partial}{\partial z} C(x, y, z) \left[v_c(x, y, t_0 + \frac{\tau}{2} - \frac{\delta}{2}) - v_c(x, y, t_0 + \frac{\tau}{2} + \frac{\delta}{2}) \right] \cos \omega_r t \end{aligned} \quad (7.8)$$

Thus, the cantilever vibration at ω_r is proportional to the change in the circuit potential over an interval δ . Unlike amplitude modulated pulse sampling, it is not clear how this relationship could be used to extract an unknown circuit signal. Figure 7.2 shows an example of the deflection that would result from sweeping t_0 from 0 to T at a fixed point (x_0, y_0, z_0) above an integrated circuit.

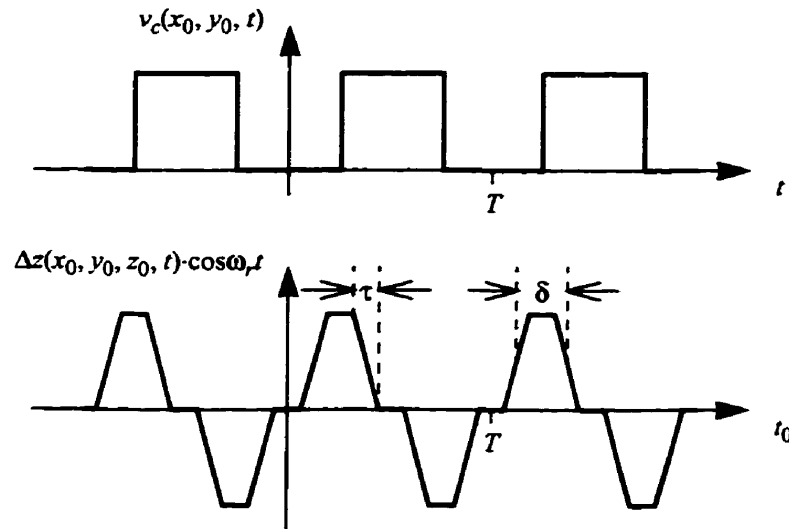


Figure 7.2 An example of actual (top) and measured circuit signals using phase modulated rectangular pulse sampling.

As expected, the cantilever deflection does not resemble the circuit signal at all. In fact, it actually resembles the derivative of the circuit signal, averaged with a rectangular window of width δ . Notice the deflection signal's rise- and fall-times. For amplitude modulated EFM with rectangular sampling pulses of width δ , the deflection signal will have a minimum transition time of δ . Phase modulation effectively samples the circuit waveform with pulses of width τ , producing a deflection signal with a minimum transition time of τ . This suggests that phase modulated rectangular pulse sampling is a better way to perform signal timing measurements than amplitude modulated rectangular pulse sampling, since the measured edges are more localized.

Phase modulated pulse sampling is best applied to a known digital signal with edges separated by more than δ , so that the effective sampling pulses only interact with one edge at a time. The technique is not suitable for measuring unknown signals since it does not provide clear information about the signal's shape. One application where it might be useful is in measuring clock skew in integrated circuits. Propagation delays can cause a clock edge to arrive at different parts of an IC at different times. This difference is called clock skew, and can cause synchronization problems and errors. As clock speeds increase, clock skew tolerances decrease. Worse, as chip dimensions shrink interconnect delays begin to dominate over gate delays, and clock skew depends more on a chip's physical layout than its logic design. This effect makes clock skew difficult to model. Since clocks are simple, well-defined signals, they are good candidates for phase modulated pulse sampling.

As in amplitude modulation, decreasing the effective sampling pulse width by decreasing τ improves temporal resolution, but reduces the cantilever deflection and degrades the instrument's signal-to-noise ratio. A limit is reached when τ equals the sampling signal transition time. At this point the leading edge of an effective sampling pulse is immediately followed by the trailing edge, and the pulse shape is more Gaussian than rectangular. This distortion should not have much impact on the measured cantilever deflection, since the pulses are only a small fraction of the effective sampling signal. The temporal resolution of these pulses can be estimated from their full width at half their height (their full-width half-maximum or FWHM dimension), which is equal to the sampling signal transition time at the limit. If τ is lowered below this value, the effective sampling pulses begin to decrease in height while their FWHM remains constant. This reduces the cantilever deflection but does not improve the temporal resolution. Thus, the temporal resolution cannot be pushed beyond the sampling signal transition time.

Phase modulated pulse sampling does not inherently provide better temporal resolution than amplitude modulated pulse sampling, since both are limited by the sampling signal transition time. However, it can have a practical advantage because it does not require that the sampling signal's transitions be placed close together, possibly making it easier to generate the sampling signal.

7.4 Fixed-point measurements

7.4.1 Experimental setup

The first phase modulated pulse sampling experiment involved the measurement of a signal at a fixed point on a microstrip transmission line. A block diagram of the experimental setup is shown in Figure 7.3.

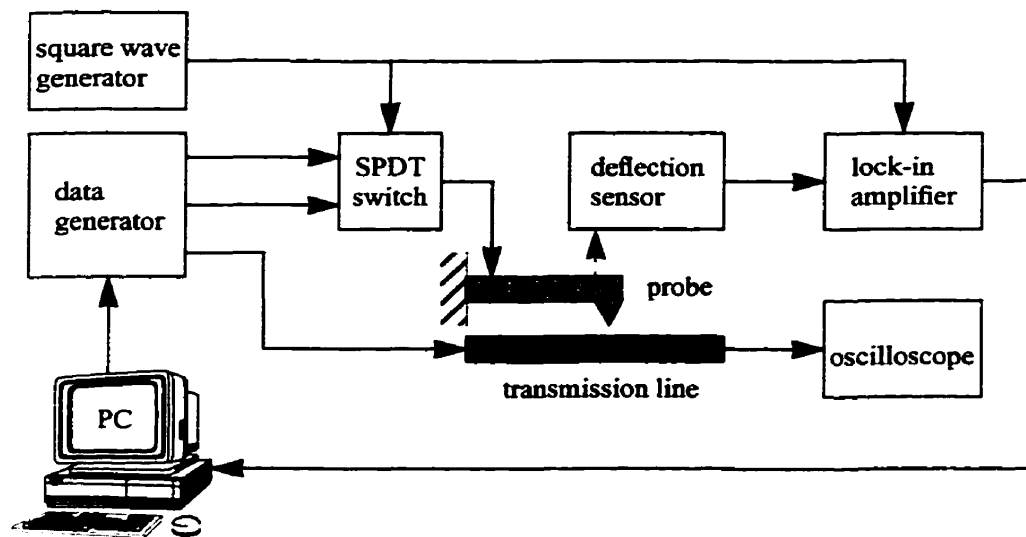


Figure 7.3 Block diagram of the microstrip transmission line experimental setup.

Both sampling and circuit signals were generated by a Hewlett-Packard HP 80000 Data Generator System. The HP E2903A 1 GHz Data Module in this system is capable of a 1 Gb/s data rate and has a typical transition time of less than 150 ps from 10% to 90% of amplitude. The delay between channels is variable with a resolution of 2 ps, and the typical rms jitter is less than 10 ps [33].

Two channels of the data generator were connected to a Mini-Circuits MSWA-2-20 single-pole double-throw absorptive switch mounted on a small printed cir-

cuit board. This switch has a bandwidth of 2 GHz, a typical switch-on time of 5.5 ns from 50% of control to 90% of RF amplitude, and a typical switch-off time of 3 ns from 50% of control to 10% of RF amplitude. The typical transition time for the RF envelope is 3 ns from 10% to 90% of amplitude [34]. A square wave generator tuned to the cantilever's resonant frequency controlled the switch. Some simple electronics were used to produce the complementary control signals with extrema of 0 and -8 V required. Any skew between the two signals resulting from the difference in path lengths to the switch was removed by monitoring the third terminal of the switch on a Tektronix 11801B Digital Sampling Oscilloscope and adjusting the data generator's deskew controls.

The switch output was then connected to a 3 dB attenuator and a short, tapered microstrip transmission line. A gold-coated SPM cantilever probe was attached to the end of the line with silver epoxy. This open-circuit termination doubles the signal at the end of the cantilever and increases the cantilever deflection. The reflected signal is absorbed by the data generator.

A laser beam-bounce system detected the cantilever vibration. The difference signal from the split photodetector was applied to a Stanford Research Systems SR510 Lock-In Amplifier, referenced to the modulating square wave. The lock-in amplifier output was maximized by adjusting the phase of its reference channel.

A third channel of the HP 80000 data generator was connected to a $50\ \Omega$ copper microstrip transmission line. This transmission line had a substrate dielectric constant of 2.5, a substrate thickness of 1.588 mm, and a strip width of 4.51 mm. The line was in turn

connected to the oscilloscope to monitor the circuit signal during the experiment and to terminate the line. A fourth data generator channel triggered the oscilloscope.

The beam-bounce cantilever deflection detection system, cantilever probe, and the transmission line under test were mounted on a rigid anodized aluminum stage, as shown in Figures 7.4 and 7.5. Micrometers allowed positioning of the cantilever deflection detection system and transmission line around the cantilever probe. A spectrum analyser, optical microscope and multimeter were on hand to help with this, and with tuning the square wave generator to the cantilever's resonant frequency. Software on a PC attached to both the data generator and lock-in amplifier was modified to manipulate the sampling and circuit signals as appropriate for phase modulated EFM and log the cantilever deflection. This software typically averages the lock-in amplifier output 1000 times over 1.25 s.

7.4.2 Results and discussion

The data generator was programmed to produce Non-Return-to-Zero (NRZ) signals at 1 Gb/s, with extrema of ± 1 V into $50\ \Omega$. The circuit signal was set to 11110000, while the two sampling signals were set to 10000000 with a 50 ps delay between them. These signals were measured with the oscilloscope and are shown in Figures 7.6 and 7.7.

The lock-in amplifier sensitivity was set to $200\ \mu\text{V}$, the time constant was set to 30 ms, and the probe flying height was estimated at $1\ \mu\text{m}$. The circuit signal was then shifted through its entire period of 8 ns in steps of 50 ps. The recorded cantilever vibrations at ω_r are shown in Figure 7.8, along with the expected results based on the measured circuit and probe signals. The expected cantilever vibration was found by calculating

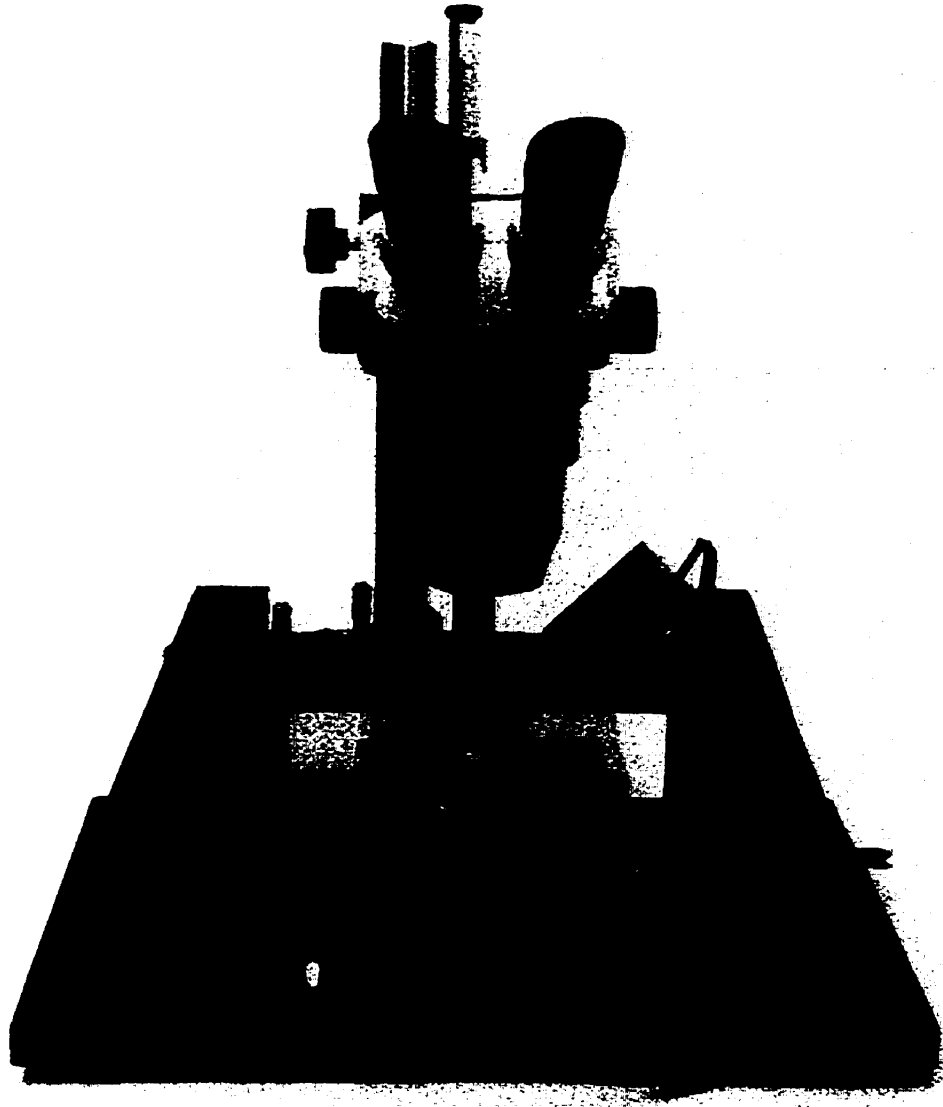


Figure 7.4 Photograph of the electrostatic force microscope and optical microscope [22].

$\int_0^T [v_s(t) - v_s(t - \tau)] v_c(x_0, y_0, t) dt$ as a function of the phase difference t_0 between the sampling and circuit signals, where $v_s(t)$ and $v_s(t - \tau)$ are the two sampling signals and $v_c(x_0, y_0, t)$ is the circuit signal. This is the only part of (7.6) that varies with t_0 . The magnitude, polarity, and phase of the measured cantilever vibrations depend on Q , k ,

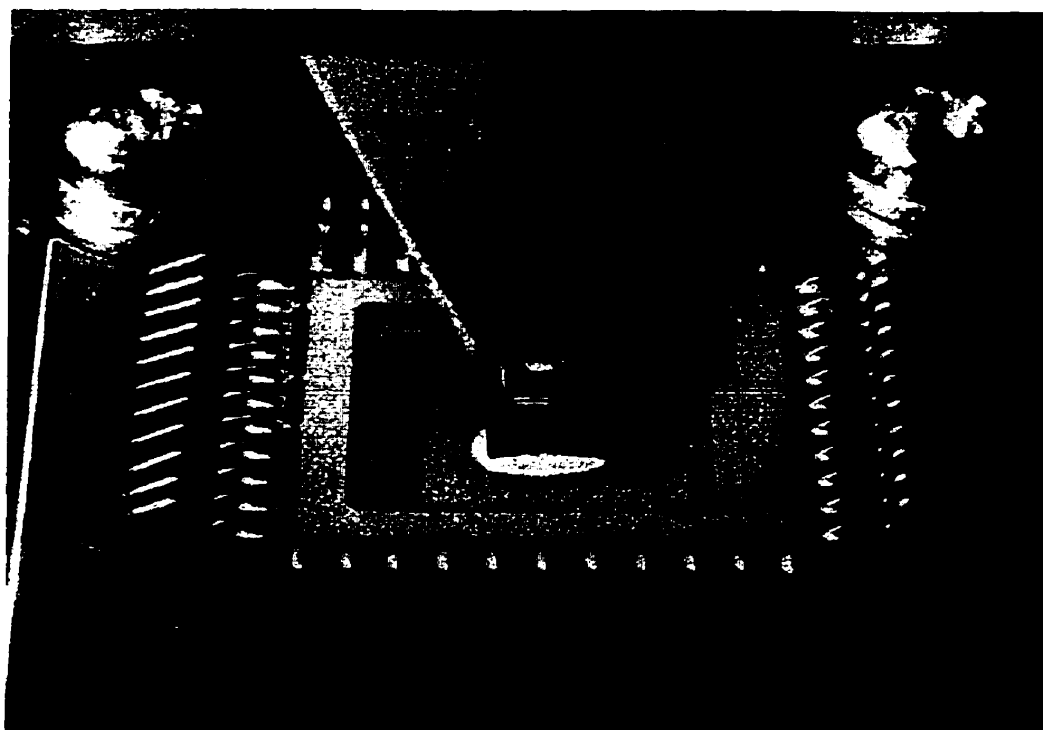


Figure 7.5 Photograph of the EFM cantilever probe transmission line (triangular ground plane facing camera) and an IC under test [22]. The probe itself is too small to be seen here.

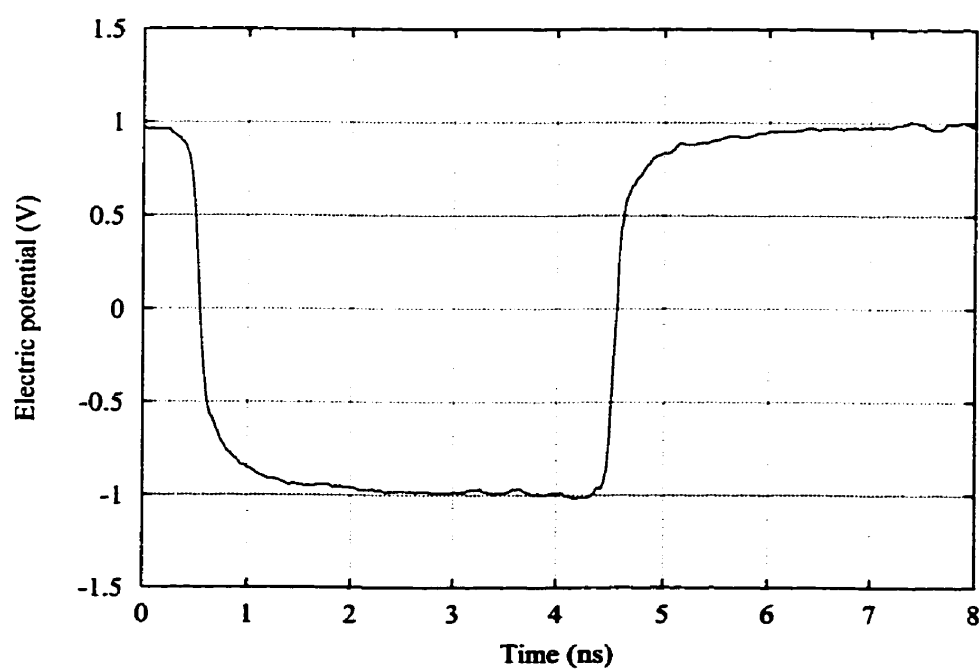


Figure 7.6 Transmission line signal measured by oscilloscope.

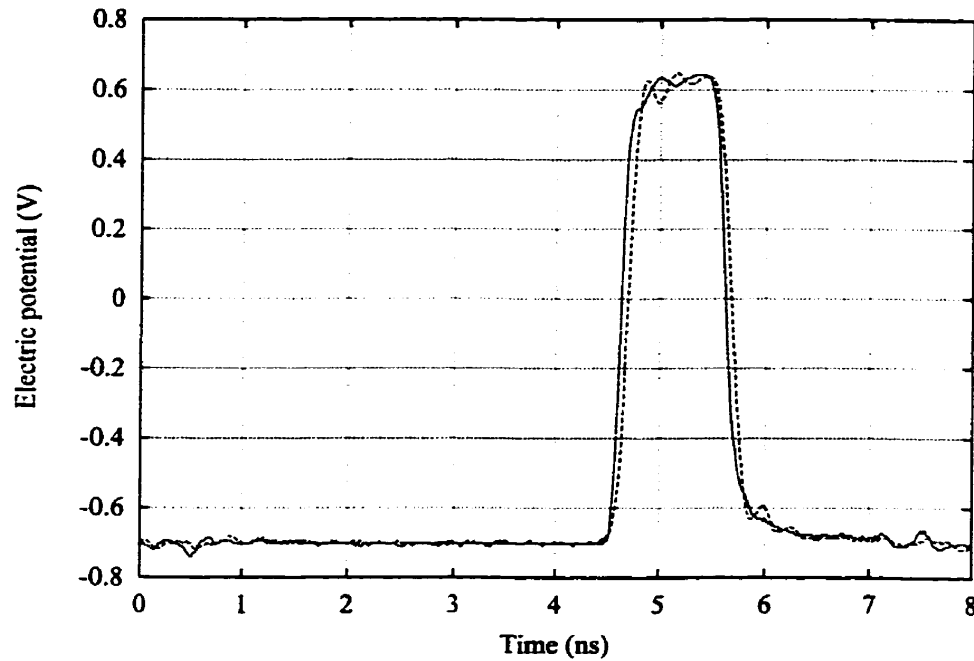


Figure 7.7 Sampling signals with a 50 ps delay measured by oscilloscope.

$\frac{\partial}{\partial z}C(x_0, y_0, z_0)$, the lock-in amplifier configuration, and the lengths of the cables that carry the circuit and probe signals, so the calculated signal was shifted and scaled to fit the data.

The measured cantilever vibration matches the expected vibration quite well, although neither vanishes between the two peaks as they would in the ideal case. This error is likely caused by the slight difference in the shape of the two sampling signals. No significant disturbance in the circuit signal was observed during the measurement.

While being deskewed at the beginning of this experiment, the shapes of the two sampling signals changed as they were shifted in time. This effect is probably caused by the switch, not the data generator.

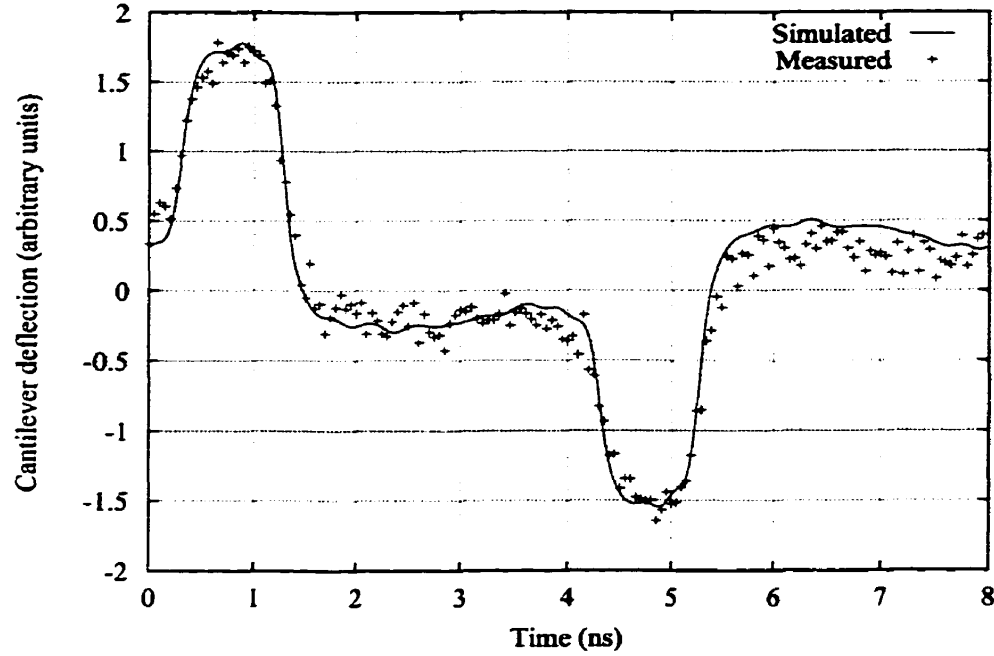


Figure 7.8 Phase modulated EFM measurements plotted against $\int_0^T [v_s(t) - v_s(t - \tau)] v_c(x_0, y_0, t) dt$.

Next, the effect of the phase modulation τ on the effective sampling signal was examined. Sampling signals similar to those in Figure 7.7 with various delays were recorded with the oscilloscope and subtracted to obtain the effective sampling signals shown in Figure 7.9. The cantilever vibrations expected from sampling an ideal 50% duty cycle square wave circuit signal with a 20 ps transition time with these signals are shown in Figure 7.10, scaled to equal amplitudes.

As τ decreases, the amplitude of the effective sampling signal decreases. This degrades the instrument's signal-to-noise ratio, which is especially noticeable in the 50 ps and 20 ps cantilever vibrations. The effective sampling signal is also distorted, with significant size differences between the two pulses in the 20 ps case. Corresponding nonideali-

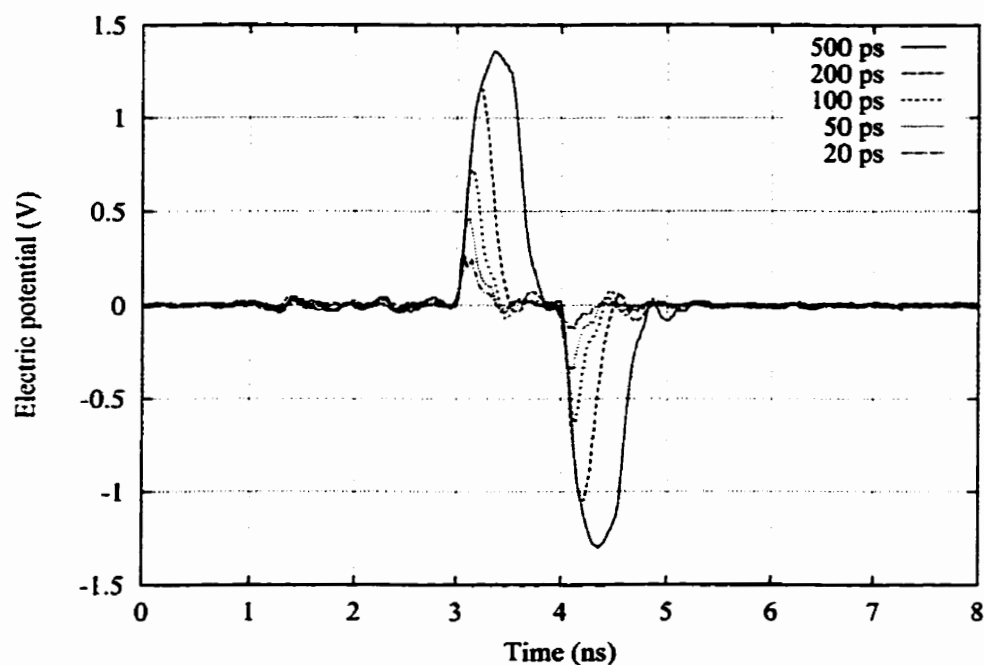


Figure 7.9 Effective sampling signals for various phase modulations.

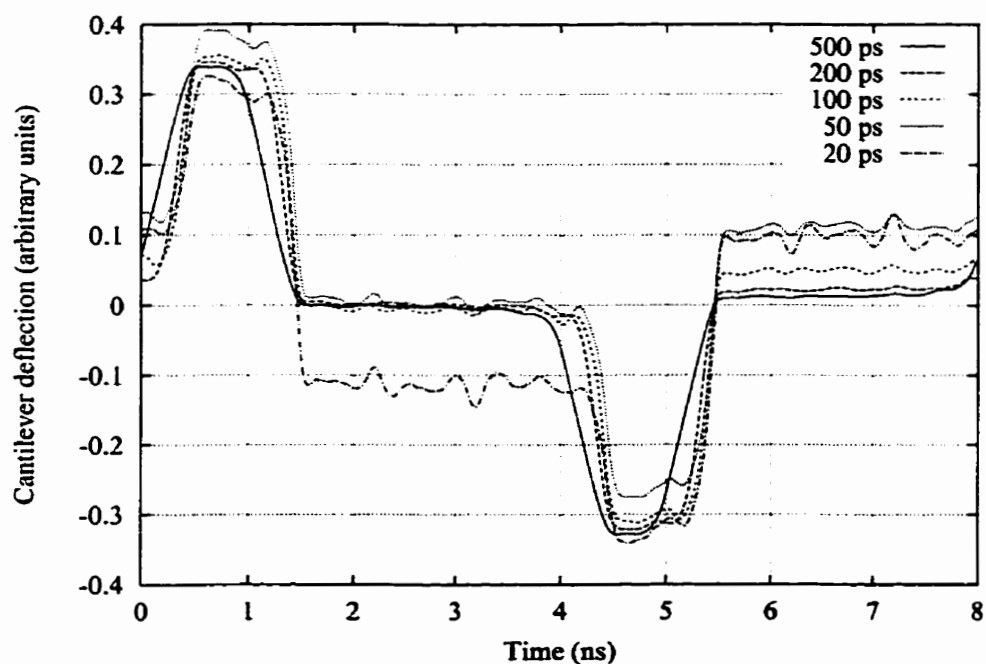


Figure 7.10 Expected cantilever vibrations for various sampling signal phase modulations.

ties can be seen in the cantilever vibration, such as the failure to vanish between the two

peaks.

The 10%-to-90% cantilever vibration transition time drops as τ is reduced from 500 ps to 200 ps, but then remains constant at about 280 ps for further decreases in τ . Since this is much greater than the 20 ps circuit signal transition time, it appears to be the temporal resolution limit for these sampling signals. As expected, it is comparable to the sampling signal transition time, which is also estimated at about 280 ps from Figure 7.7. This is greater than the specified transition time of the data generator, and is probably an effect of the switch's finite bandwidth.

Finally, some propagation delay measurements were made on the transmission line. The same circuit and sampling patterns were used, with a 100 ps delay between the two sampling signals. This time, the circuit signal was shifted 1 ns in steps of 2 ps as attention was focused on one transition in the cantilever vibration. This procedure was repeated six times as the probe was moved along the length of the transmission line in 2 mm increments. The resulting cantilever deflection signals were smoothed with a nine-point (18 ps) rectangular window and are plotted in Figure 7.11.

Assuming the strip thickness is negligible, the effective dielectric constant of a microstrip transmission line is

$$\epsilon_{re} = \frac{\epsilon_r + 1}{2} + \frac{\epsilon_r - 1}{2} \frac{1}{\sqrt{1 + 10h/W}} \quad (7.9)$$

where ϵ_r is the substrate dielectric constant, h is the substrate thickness, and W is the strip width [35]. The phase velocity is

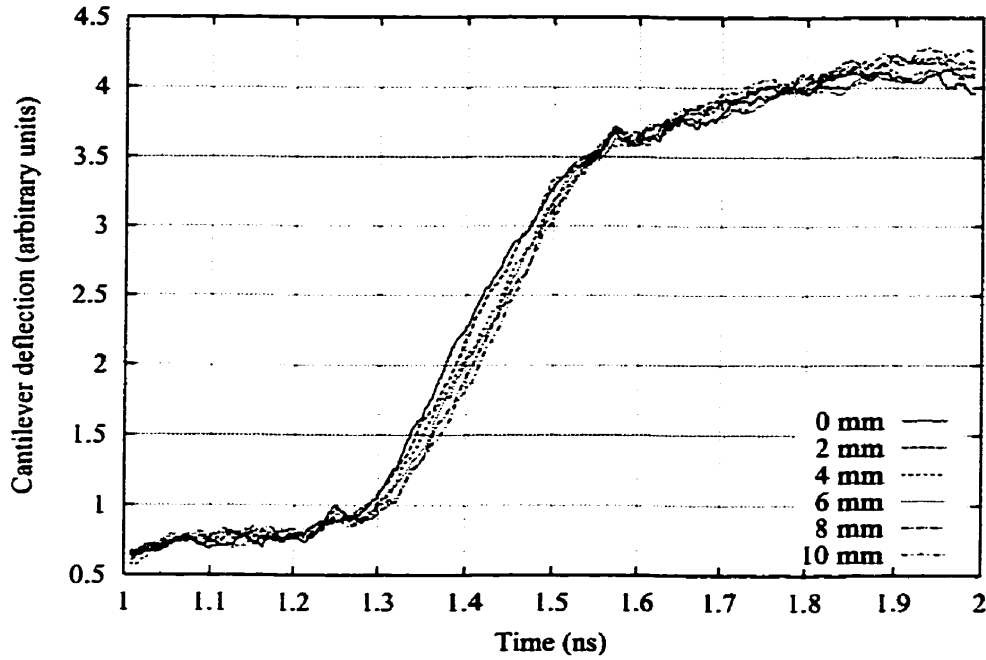


Figure 7.11 Transmission line propagation delay measurements, smoothed with an 18 ps moving average.

$$v_p = \frac{c}{\sqrt{\epsilon_{re}}} \quad (7.10)$$

where c is the speed of light in vacuum. The propagation delay for any line length d can then be calculated as

$$t = \frac{d}{v_p}. \quad (7.11)$$

For a 2 mm length of this line, the propagation delay should be 9.68 ps.

Figure 7.12 shows straight lines fitted to the transitions in each raw data set, between cantilever deflections of 1.5 and 3.0. The time at which each of these lines crosses a cantilever deflection of 2.25 is plotted in Figure 7.13, along with their standard errors. A line in turn fitted to these times gives a propagation delay of 6.71 ± 0.395 ps for a 2 mm length of transmission line.

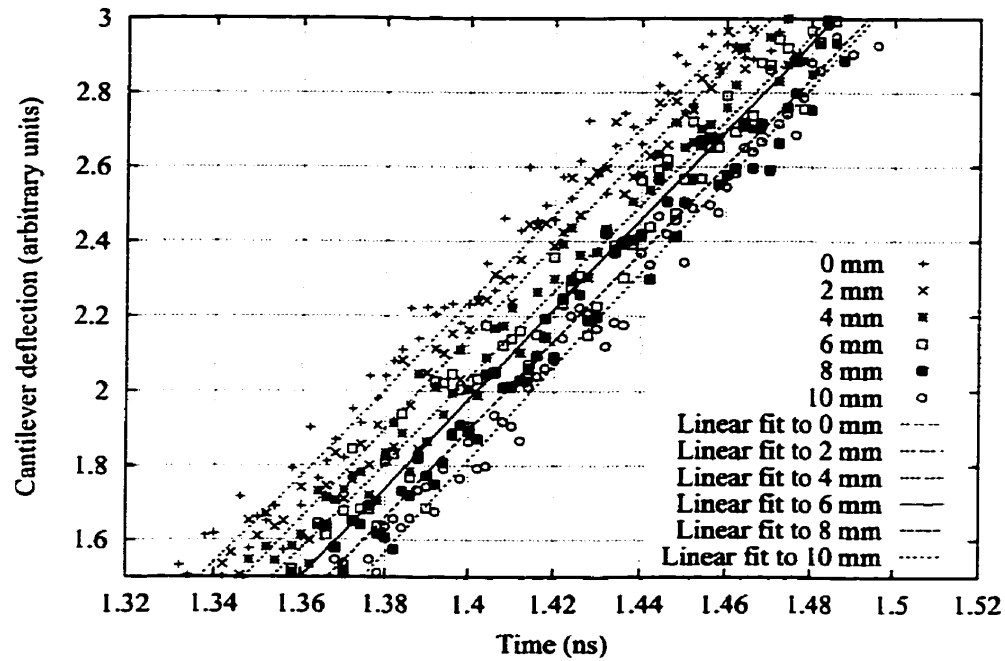


Figure 7.12 Straight line fits to the transitions in EFM propagation delay measurements.

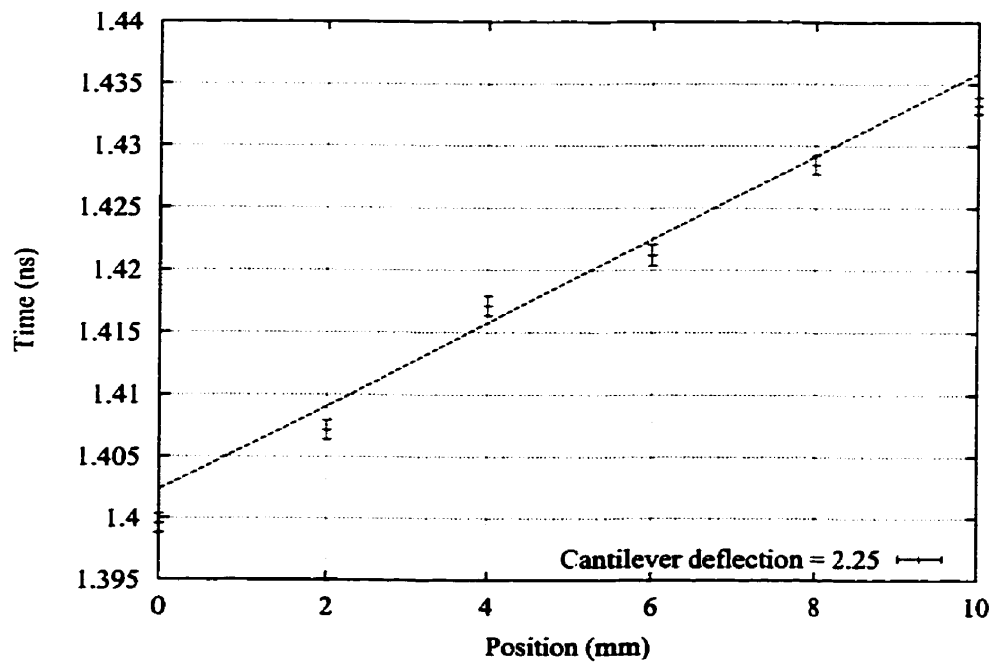


Figure 7.13 Propagation delays along a microstrip transmission line.

The measured propagation delay is much smaller than the expected delay. The fact

that the fitted line in Figure 7.13 lies outside most of the error bars suggests that there is a systematic error in the measurement which changes from position to position. One source of this error may be uncertainty in the relative positions of the probe and transmission line. It is difficult to orient the line so that it is parallel with one of the micrometer axes. Also, the probe-sample distance is set with a tripod of micrometers. Both factors make it awkward to move the probe a known distance along the length of the line while maintaining a constant flying height.

7.5 One-dimensional scanned measurements

7.5.1 Experimental setup

The purpose of this experiment was to use phase modulated EFM to measure the gate delay of an inverter and to test the lift mode of the digital SPM controller. The device under test was an integrated circuit designed for internal node probing [36]. This IC features a simple layout and packaging that allows easy access to a set of unpassivated test pads, although these were not used in this experiment. A photograph of the IC is shown in Figure 7.14.

A piezoelectric scanner was added to the EFM stage, allowing the controller to manipulate the probe. The probe was positioned over the tight cluster of interconnects running from left to right on the left side of Figure 7.14. These interconnects tap the outputs of a chain of cascaded CMOS inverters and carry them to a set of test pads. Figure 7.15 shows a schematic of the inverter chain and the taps.



Figure 7.14 Photograph of the cantilever probe and integrated circuit under test [22].

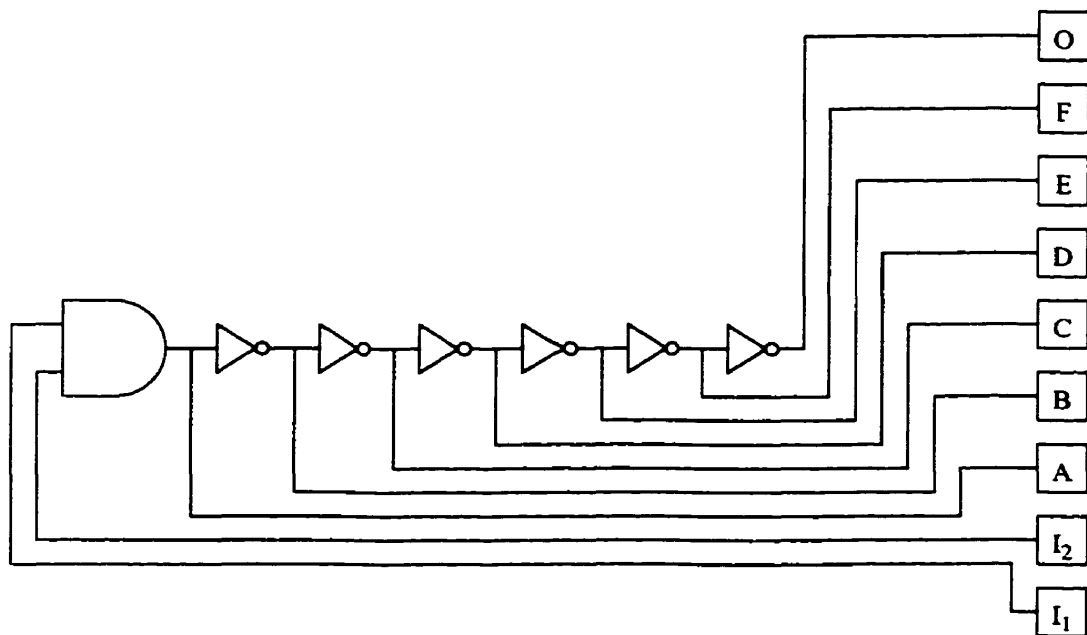


Figure 7.15 Diagram of the inverter chain and test pads.

The piezoelectric scanner has its own high-voltage driver, so the digital SPM controller's high-voltage outputs were connected to the driver's external control inputs through a voltage divider. This driver is capable of closed loop operation using position sensors built into the scanner, but the loop was disabled because the control system occasionally becomes unstable. The cantilever deflection signal and the output of the lock-in

amplifier were connected to the controller to complete the AFM feedback loop and to allow the controller to record the cantilever deflection at resonance. Unlike the PC used for the fixed-point measurements, the digital SPM controller does not average the lock-in amplifier output.

Besides the test pads labelled I_1 and I_2 in Figure 7.15, the inputs of the inverter chain are connected to I/O pads that allow them to be driven by external signals applied to pins on the package of the integrated circuit. The IC was mounted on a small printed circuit board that joins one of these pins to a standard connector with a $50\ \Omega$ transmission line, and ties the other to the 5 V dc supply powering the chip. The data generator was programmed with the same circuit and sampling patterns as in the fixed-point case, with extrema of 0 V and 2.5 V into $50\ \Omega$ and a 100 ps delay between the two sampling signals. The circuit signal drove the integrated circuit input.

7.5.2 Results and discussion

The digital SPM controller performed a $40\ \mu\text{m}$ linear scan across interconnects D, E and F at a rate of 0.5 Hz, with a lift height of 600 nm. The time constant of the lock-in amplifier was set to 10 ms, which at a scan frequency of 0.5 Hz is about 2.5 pixels or 400 nm. After every sixteenth sweep of the scanner, the circuit signal was shifted by 100 ps. This produced images of the circuit topography and electrostatic force versus position and time, as shown in Figure 7.16.

Since the digital SPM controller performs 256 raster lines per image, only 1.6 ns of the 8 ns circuit signal period was measured. However, this was enough to capture a signal edge as it moved through the inverter chain. The inversion of the signal between intercon-

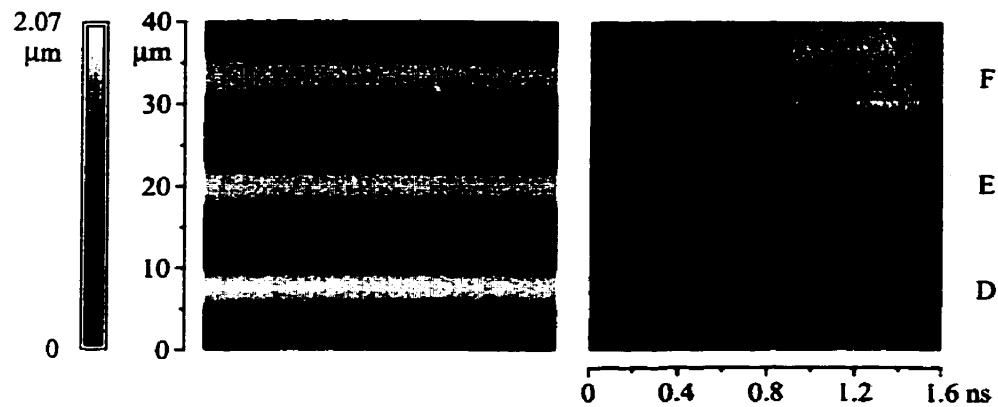


Figure 7.16 Linear topographic (left) and electrostatic force measurements versus time.

nects is clear as the electrostatic force image switches between light and dark, representing reversals in the phase of the cantilever vibration.

This electrostatic force image is somewhat misleading, since instead of flowing continuously from 0 ns to 1.6 ns the time axis actually makes discrete steps of 100 ps. Averaging the data over each 100 ps interval gives a more accurate representation. Subsequently averaging a region about 5 μm wide above each interconnect creates the cantilever deflection signals plotted in Figure 7.17.

The small number of data points collected makes it inappropriate to calculate the correlation coefficient in this case. The distortion in the cantilever deflection signals is substantial, but the gate delay of each inverter can be estimated at 200 ps. Given that the data points are 100 ps apart, this estimate is in reasonable agreement with previous measurements and models of this circuit, which gave values of 150 ps and 190 ps respectively [22].

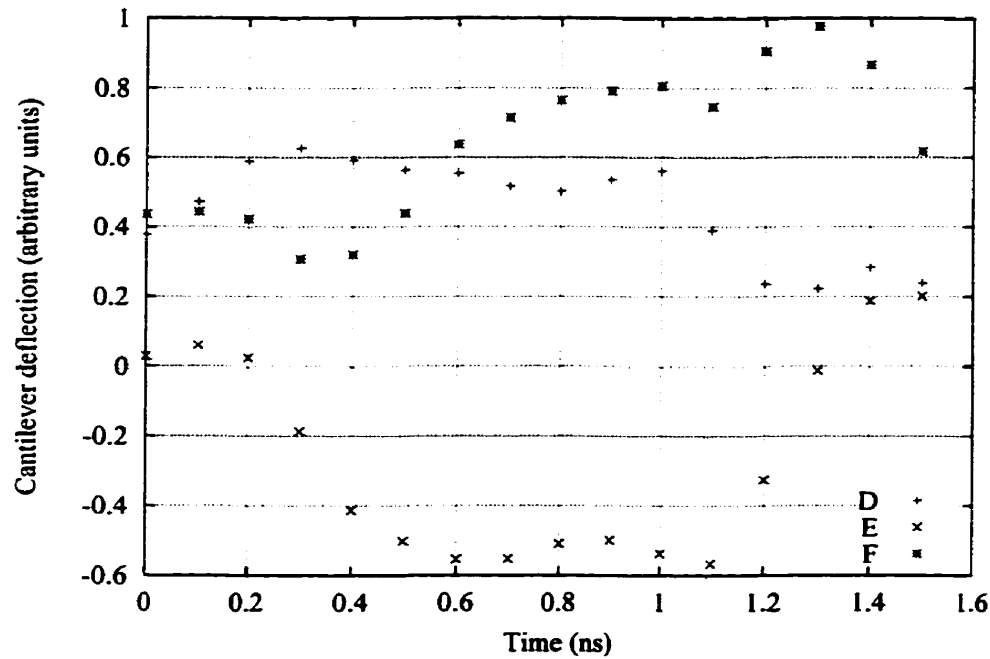


Figure 7.17 Average phase modulated EFM measurements for each interconnect.

Figure 7.18 shows cross-sections of the average topographic and electrostatic force data during the 0.8 ns interval. The estimated path of the probe during lift mode is also included.

There are a number of observations that can be made from this figure. First, there seems to be some distortion in the lateral axis of the topography data. These interconnects are 10 μm apart, but appear about 13 μm apart in the measured topography. This distortion is likely caused by miscalibration of the piezoelectric scanner, since it is not operating in its intended closed loop fashion.

The spatial resolution of the electrostatic force image seems quite poor. This problem is probably due to the protective passivation layer that covers the interconnects and increases the distance between them and the probe. Although the thickness of this layer is proprietary information, it can be estimated by comparing the measured topography with

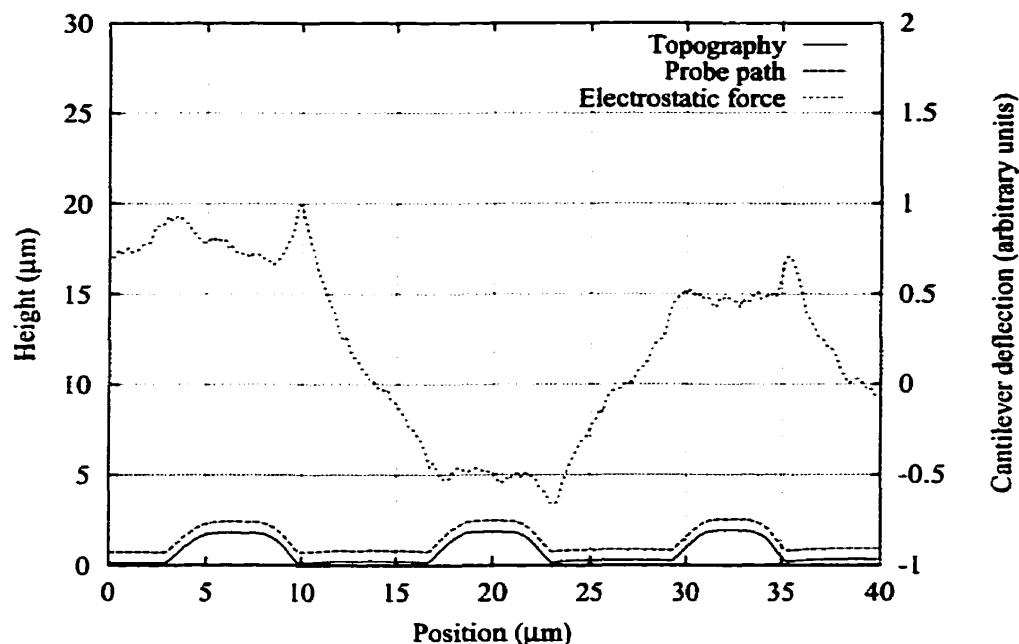


Figure 7.18 Average topographic and electrostatic force signals during the 0.8 ns interval.

the design. These interconnects are $1\text{ }\mu\text{m}$ wide, but their full-width half-maximum dimension in the topographic image is about $4.7\text{ }\mu\text{m}$. Assuming that this overstates their true width by about thirty percent as suggested above, the interconnects are about $3.6\text{ }\mu\text{m}$ wide. Thus, the surrounding passivation is estimated at about $1.3\text{ }\mu\text{m}$ in thickness, meaning the actual probe flying height is almost $2\text{ }\mu\text{m}$. As previously discussed, an increase in the flying height severely impairs the spatial resolution.

Finally, the electrostatic force signal strengthens at the sidewalls of the interconnects. This enhancement is believed to result from an increase in the magnitude of the capacitance gradient here. The effect is asymmetric, with larger peaks at the trailing edges of the lift mode pass. This is probably caused by a tilt between the probe and the circuit.

While scanning, a squeaking or chirping noise was heard each time the piezoelectric scanner withdrew the probe to acquire a $z(z)$ curve at the start of a lift mode line. This

sound is attributed to the coarse stepwise movement of the scanner when it retracts the probe, since similar movements with finer steps do not generate any noise. The effect is believed to be harmless.

Although outputs A, C, E and O are available at external pins, the circuit is unable to drive large off-chip loads. Thus, the invasiveness of this measurement could not be tested.

In this experiment the circuit signal phase was shifted manually after every sixteenth lift mode pass, making the measurement fairly labour-intensive. The PC that controlled the data generator in the fixed-point measurement has also been used to shift the circuit signal phase during scanned measurements. However, since this PC is not synchronized with the digital SPM controller, the phase sometimes shifts during a lift mode pass. This causes short streaks to appear along the fast-axis direction in the electrostatic force image. In future experiments, some way of synchronizing the data generator with the digital SPM controller should be found. A user interface that is not restricted to 256 raster lines would also be helpful.

7.6 Two-dimensional scanned measurements

7.6.1 Experimental setup

The purpose of this experiment was to obtain a series of two-dimensional phase modulated EFM images of an integrated circuit at various phases. These images would be snapshots of the electric potential distribution across the circuit at different instants in time, and could be used to track a signal as it propagates through the circuit. The same IC

was used as in the one-dimensional scanned measurement, and the setup was nearly identical.

7.6.2 Results and discussion

The probe was positioned directly over the inverter chain, and a $40\text{ }\mu\text{m}$ by $40\text{ }\mu\text{m}$ area was scanned at 0.5 Hz with an 800 nm lift height. The circuit and sampling signals were as before, with the sampling signals separated by 100 ps . The phase of the circuit signal was set so that a transition would be captured within the scanned area. The resulting topographic and electrostatic force images are shown in Figure 7.19.



Figure 7.19 Topographic (left) and electrostatic force images of the inverter chain.

The electrostatic force image appears much as expected, with consecutive interconnects alternating between light and dark. The scanner miscalibration and electrostatic force enhancement at the interconnect sidewalls seen in the previous experiment are also present. Unfortunately, attempts to obtain additional images at different phases were unsuccessful. The cantilever deflection signal fluctuated wildly at times, especially at the interconnect sidewalls. A small amount of this can be seen at the bottom of the electrostatic force image. This fluctuation is probably due to the probe contacting the surface during lift mode, which is more likely to happen at an abrupt change in topography.

This experiment was performed before linear interpolation of the scanner position between image points in lift mode was added. This change would make the lift mode scan smoother and could fix this interconnect sidewall problem. It is also possible that there is some long-term charge buildup in or on the passivation layer that causes it to attract the probe. In this case, scanning an unpassivated region may be helpful. Scanning a planarized circuit would almost certainly solve the problem. However, these last two suggestions involve changing the structure of the integrated circuit, which may not be possible or desirable.

7.7 Suggestions for future work

Phase modulated electrostatic force microscopy is a simple technique that allows some improvement in the temporal resolution of EFM measurements that use rectangular sampling pulses. In exchange, it sacrifices information about the shape of the measured signal. An alternative to using rectangular sampling pulses is to use sawtooth sampling signals. The resulting effective sampling signal is a train of unipolar rectangular pulses similar to those used in amplitude modulated pulse sampling, as shown in Figure 7.20. The effective sampling signal does not vanish between pulses, so there will be some distortion in the measured circuit signal, but it should still resemble the actual circuit signal. This sampling signal must only have a single fast transition, instead of two in opposite directions spaced closely together.

The most important idea to come out of this study of phase modulated EFM may be that modulation schemes other than amplitude modulation can be used. In a communi-

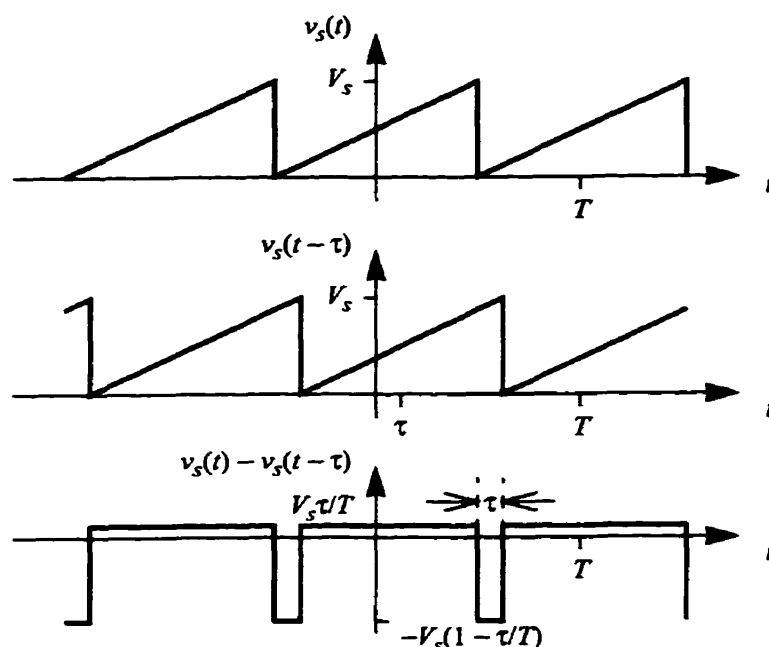


Figure 7.20 Phase modulated sawtooth sampling signals (top, middle) and the equivalent amplitude modulated sampling signal.

cation system, the phase modulation described here would be called pulse position modulation. Pulse position modulation is one of two main types of pulse time modulation. The other is pulse width modulation, and is shown in Figure 7.21. Pulse width modulation could achieve the same temporal resolution gains as phase modulation without losing the measured signal shape information. The mathematics of this scheme are slightly different from phase modulation since the two sampling signals do not have the same shape, but the equivalent amplitude modulated sampling signal should again be a train of narrow unipolar pulses.

While the data generator used in this work cannot adjust bit widths in increments smaller than 1 ns, smaller pulse width modulations can be implemented by adding a second switch to the phase modulation equipment. As shown in Figure 7.22, this switch must be fast enough to gate through one of the effective sampling pulses while blocking the

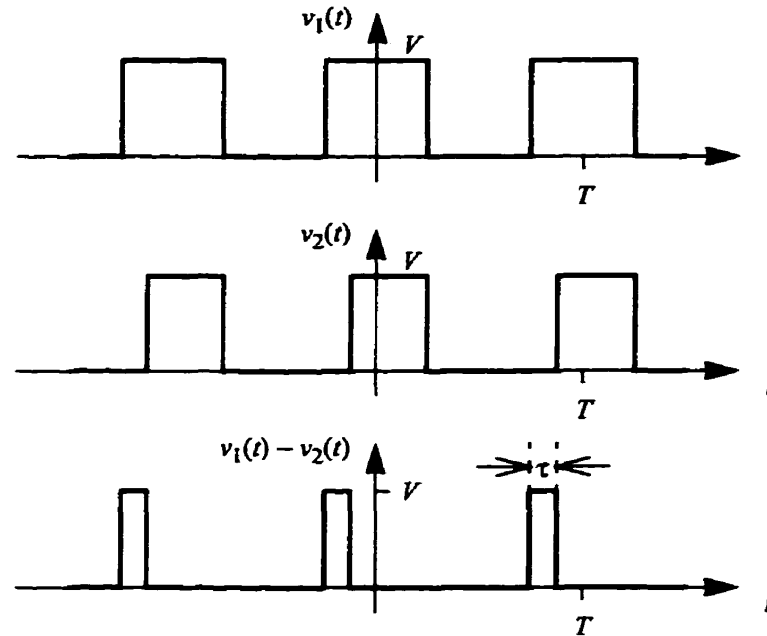


Figure 7.21 Equivalent pulse width modulated (top, middle) and amplitude modulated sampling signals. other. The result is a pulse width modulated signal where the modulation is the delay between the two sampling signals produced by the data generator.

Some scanning-related issues should also be investigated. Although the gold-coated cantilever probes used in this research perform well in fixed-point EFM measurements, the AFM portion of the lift mode algorithm could damage the gold coating during scanned measurements. The coating could be rubbed off by contact with the circuit, or it could crack when the cantilever is flexed.

The scanned measurements in this work were performed exclusively over passivated regions of an integrated circuit. However, bare metal could be encountered while scanning some circuits. The risk of damage to an IC that electrically contacts the EFM probe should be evaluated. If there is a significant risk, the probe may need to be isolated from the EFM electronics during the AFM portion of lift mode.

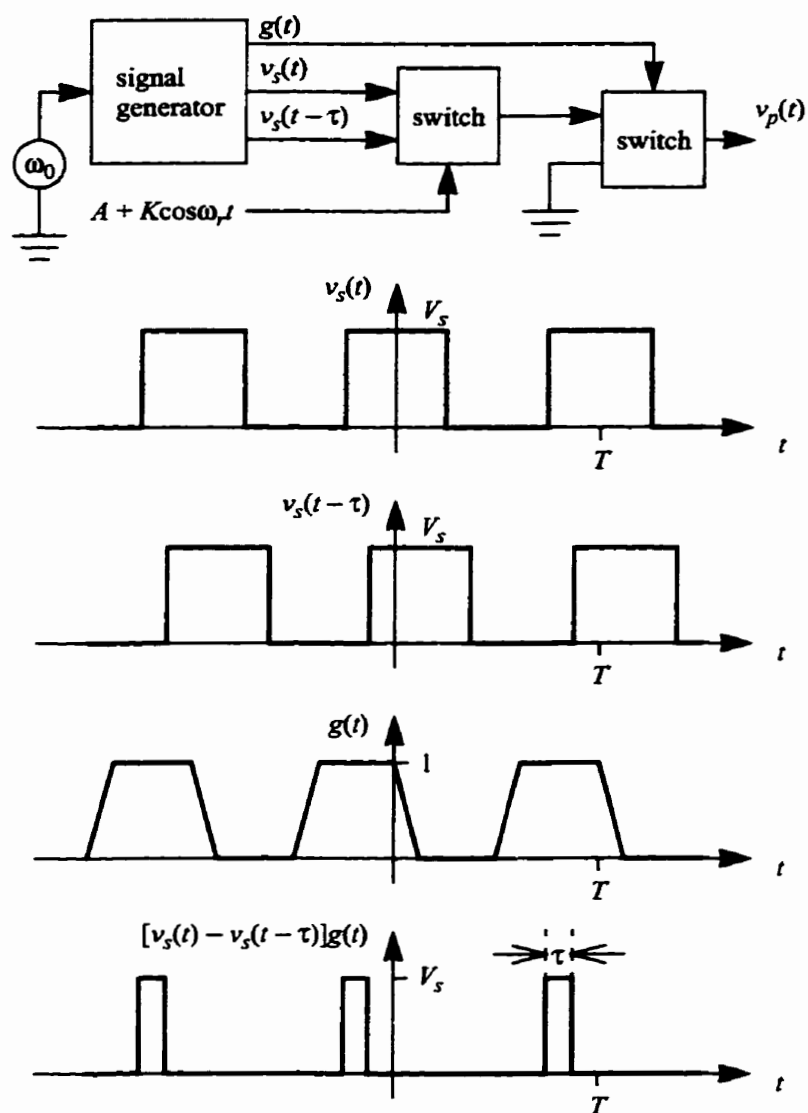


Figure 7.22 Pulse width modulation using modified phase modulation equipment.

8

Conclusions

The revised digital SPM controller software makes full use of the upgraded hardware. Several bugs were fixed, and new features such as the dynamic voltage clamp to prevent the piezoelectric scanner from depoling were added. The user interface now has TIFF data storage, an improved “look-and-feel,” and easy accommodation of different SPMs. The lift mode algorithm and second data channel allow SRM and EFM control with simultaneous topography measurement and no change in AFM performance. Sample images show the separation of topography and electrostatic force in lift mode.

The deflection of an EFM cantilever carrying a signal that is phase modulated by a 50% duty cycle square wave is the same as that produced by a related amplitude modulated probe signal. Sampling with rectangular pulses modulated in this way is equivalent to sampling with a pair of amplitude modulated pulses whose widths are limited by the amount of phase modulation and the sampling signal transition time. For a given sampling

signal, this technique can provide better temporal resolution than amplitude modulation while sacrificing information about the measured signal's shape.

This EFM technique was used to characterize transition times with 280 ps resolution, and propagation delays with 10 ps resolution on a microstrip transmission line using 1 ns sampling pulses. The skew between three integrated circuit interconnects was measured by scanning across them in the lift mode of the digital SPM controller. The controller also imaged the electrostatic force above a small area of the integrated circuit at an instant in time.

References

- [1] A. Lemus, "Design and Implementation of an All Digital Scanned Probe Microscope Controller," Master's thesis, University of Manitoba, 1995.
- [2] D. Winkler, R. Schmitt, M. Brunner, and B. Lischke, "Flexible picosecond probing of integrated circuits with chopped electron beams," *IBM Journal of Research and Development* **34** (2/3): 189-203 (1990).
- [3] P. Fazekas, H.-P. Feuerbaum, and E. Wolfgang, "Scanning electron beam probes VLSI chips," *Electronics* **54** (14): 105-112 (1981).
- [4] R. Clauberg, H. Beha, A. Blacha, and H. K. Seitz, "Picosecond photoemission probing of integrated circuits: capabilities, limitations, and applications," *IBM Journal of Research and Development* **34** (2/3): 173-187 (1990).
- [5] K. J. Weingarten, M. J. W. Rodwell, and D. M. Bloom, "Picosecond optical sampling of GaAs integrated circuits," *IEEE Journal of Quantum Electronics* **24** (2): 198-220 (1988).
- [6] J. Kim, S. Williamson, J. Ness, S. Wakana, and J. Whitaker, "Photoconductive sampling probe with 2.3-ps temporal resolution and 4-mV sensitivity," *Applied Physics Letters* **62** (18): 2268-2270 (1993).

- [7] Y. Gao and I. Wolff, "A simple electric near field probe for microwave circuit diagnostics," *1996 IEEE MTT-S International Microwave Symposium Digest*, 1537-1540 (1996).
- [8] G. Binnig, H. Rohrer, Ch. Gerber, and E. Weibel, "Surface studies by scanning tunneling microscopy," *Physical Review Letters* **49** (1): 57-61 (1982).
- [9] G. Binnig, C. F. Quate, and Ch. Gerber, "Atomic force microscope," *Physical Review Letters* **56** (9): 930-933 (1986).
- [10] R. C. Barrett and C. F. Quate, "Charge storage in a nitride-oxide-silicon medium by scanning capacitance microscopy," *Journal of Applied Physics* **70** (5): 2725-2733 (1991).
- [11] C. Shafai, D. J. Thomson, and M. Simard-Normandin, "Two-dimensional delineation of semiconductor doping by scanning resistance microscopy," *Journal of Vacuum Science and Technology B* **12** (1): 1-5 (1994).
- [12] Y. Martin, C. C. Williams, and H. K. Wickramasinghe, "Atomic force microscope-force mapping and profiling on a sub 100-Å scale," *Journal of Applied Physics* **61** (10): 4723-4729 (1987).
- [13] J. E. Stern, B. D. Terris, H. J. Mamin, and D. Rugar, "Deposition and imaging of localized charge on insulator surfaces using a force microscope," *Applied Physics Letters* **53** (26): 2717-2719 (1988).
- [14] Y. Martin, D. Rugar, H. K. Wickramasinghe, "High-resolution magnetic imaging of domains in TbFe by force microscopy," *Applied Physics Letters* **52** (3): 244-246 (1988).
- [15] R. A. Said, "Scanning Force Potentiometry Techniques for Semiconductor Circuit Characterization," Ph.D. thesis, University of Manitoba, 1996.
- [16] G. Meyer and N. M. Amer, "Novel optical approach to atomic force microscopy," *Applied Physics Letters* **53** (24): 2400-2402 (1988).
- [17] M. Nonnenmacher, M. P. O'Boyle, and H. K. Wickramasinghe, "Kelvin probe force microscopy," *Applied Physics Letters* **58** (25): 2921-2923 (1991).
- [18] J. M. R. Weaver and D. W. Abraham, "High resolution atomic force microscopy potentiometry," *Journal of Vacuum Science and Technology B* **9** (3): 1559-1561 (1991).

- [19] A. S. Hou, F. Ho, and D. M. Bloom, "Picosecond electrical sampling using a scanning force microscope," *Electronics Letters* **28** (25): 2302-2303 (1992).
- [20] G. E. Bridges, R. A. Said, and D. J. Thompson, "Heterodyne electrostatic force microscopy for non-contact high frequency integrated circuit measurement," *Electronics Letters* **29** (16): 1448-1449 (1993).
- [21] U. Mueller, C. Boehm, J. Sprengel, C. Roths, E. Kubalek, and A. Beyer, "Geometrical and voltage resolution of electrical sampling scanning force microscopy," *1994 IEEE MTT-S International Microwave Symposium Digest*, 1005-1008 (1994).
- [22] D. Noruttun, "Non-Contact Internal Probing of High Speed Microelectronic Circuits using Electrostatic Force Microscopy," Master's thesis, University of Manitoba, 1997.
- [23] G. E. Bridges and D. J. Thomson, "High-frequency circuit characterization using the AFM as a reactive near-field probe," *Ultramicroscopy* **42**: 321-328 (1992).
- [24] H. Yokoyama and T. Inoue, "Scanning Maxwell stress microscope for nanometre-scale surface electrostatic imaging of thin films," *Thin Solid Films* **242**: 33-39 (1994).
- [25] K. Yackoboski, "Design and Implementation of an All-Digital Atomic Force Microscope Controller," Master's thesis, University of Manitoba, 1992.
- [26] J. N. Nxumalo, D. T. Shimizu, D. J. Thomson, and M. Simard-Normadin, "High-resolution cross-sectional imaging of MOSFET's by scanning resistance microscopy," *IEEE Electron Device Letters* **18** (2): 71-73 (1997).
- [27] Borland Pascal with Objects 7.0, Borland International, Scotts Valley, California, 1992.
- [28] BGI256 3.02, Knight Software, Milwaukie, Oregon, 1993 (<ftp://ftp.inprise.com/pub/pascal/devsupport/general/bgi256.zip>).
- [29] Aldus Corporation, *TIFF Revision 6.0* (Seattle, Washington: Aldus Corporation, 1992) (<http://www.adobe.com/supportservice/devrelations/PDFS/TN/TIFF6.pdf>).
- [30] J. Bradley, "John's World of XV 'n' Stuff" (<http://www.trilon.com/xv/>).
- [31] Borland International, *Borland Pascal with Objects Version 7.0 Language Guide* (Scotts Valley, California: Borland International, 1992), 303.

- [32] "The NCSA HDF Home Page," National Center for Supercomputing Applications (<http://hdf.ncsa.uiuc.edu/>).
- [33] Hewlett-Packard GmbH, *HP 80000 Data Generator System Installation Guide*, Edition 1.2 (Germany: Hewlett-Packard GmbH, 1993), 2:17.
- [34] "Switches," Mini-Circuits (<http://www.minicircuits.com/npswitch.htm>).
- [35] K. C. Gupta, R. Garg, and R. Chadha, *Computer-Aided Design of Microwave Circuits* (Dedham, Massachusetts: Artech House, 1981), 61-62.
- [36] H. Pollitt-Smith, "BiCMOS Chip Fabrication Report," University of Manitoba, 1996.

Appendix A

DSP program source code

```
/* XHANDLER.C */

/* DSP56001 program */
/* Compile with "dsp56kcc xhandler" */
/* Revised by AVID LEMUS, October 1995 */
/* Revised by David T. Shimizu, June 1998 */
/* z-out and In4 are sent to the PC */
/* Reads Vectored interrupts from the HOST (PC) and multiplies the */
/* Value by 2 to get the address of the service routine to which the */
/* the program counter goes. A maximum of 32 such interrupts are */
/* available (ie. 5bits) */
/* When Doing feedback. Every time it finishes a row it should send */
/* it to the PC via an INT 7 RREQ. */
/* Now driven by timer interrupts. This program sends the line number */
/* before the data. */
/* The DSP board's RAM is partitioned as the factory default; */
/* P:$1000 X:$0800 Y:$0800 */
/* The default starting address for the main program is P:$0040 */
/* Interrupt vectors reside at P:$0000 - P:$003e */

/* DSP constants CONTROL REGISTERS and such */

#define PBC X:$ffe0 /* Port B Control Register */
#define PCC X:$ffe1 /* Port C Control Register */
#define PCDDR X:$ffe3 /* Port C Data Direction Register */
#define PCD X:$ffe5 /* Port C Data Register */
#define HCR X:$ffe8 /* Host Control Register */
#define HSR X:$ffe9 /* Host Status Register */
#define HRX X:$ffeb /* Host Receive Data Register */
#define HTX X:$ffeb /* Host Transmit Data Register */
#define SCR X:$fff0 /* Serial Comm. Interface Control Register */
```

```

#define SCCR X:$fff2    /* SCI Clock Control Register */
#define BCR X:$fffe    /* Bus Control Register */
#define IPR X:$ffff    /* Interrupt Priority Register */

/* Memory mapped I/O constants...(i.e. Analog I/O board) */

#define CDA1 Y:$a000    /* X D/A, 20 bits, +/- 200 V */
#define CDA2 Y:$a001    /* Y D/A, 20 bits, +/- 200 V */
#define CDA3 Y:$a002    /* Z D/A, 20 bits, +/- 200 V */
#define CDA4 Y:$a003    /* V D/A, 20 bits, +/- 10 V */
#define CDA5 Y:$a004    /* V D/A, 20 bits, +/- 10 V */
#define CDA6 Y:$a005    /* V D/A, 20 bits, +/- 10 V */
#define FDPF Y:$a006    /* Digitally Programmable Lowpass Filter */

#define CAD1 Y:$a000    /* In 1 A/D, 18 bits, Convert 1&2 */
#define CAD2 Y:$a001    /* In 2 A/D, 18 bits */
#define CAD3 Y:$a002    /* In 3 A/D, 18 bits, Convert 3&4 */
#define CAD4 Y:$a003    /* In 4 A/D, 18 bits */
#define CAD5 Y:$a004    /* In 5 A/D, 18 bits, Convert 5&6 */
#define CAD6 Y:$a005    /* In 6 A/D, 18 bits */

/* Constants */
/* The DSP56001 has a clock speed of 20.48 MHz */
/* Therefore one instruction cycle is 97.7 ns */
/* REP cannot use immediate data greater than 4095 */

#define ADC_CONV 66      /* ADC conversion time = 6.4 us */
#define MOTOR_STEP 10240 /* one motor step = 1 ms */
#define ZZ_STEP 4094     /* z(z) curve sampling period = 400 us */
#define LIFT_STEP 2047   /* lift mode z(z) curve sampling period = 200 us */
#define CD 9             /* Clock divider, 0 to 4095 */
                        /* SCP = 0 */
                        /* SCI timer period = 64 * (7 * SCP + 1) * (CD + 1) / 20.48 MHz */
                        /* = 31.25 us */
#define FEED_DELAY 3200 /* number of timer periods to wait */
                        /* before starting lift mode (100 ms) */

/* Some Variables used in this program */

int darray[512];        /* new image data */
int i;                  /* counter */
int datacounter = 0;     /* index to data array */
int xoffset = 0, yoffset = 0; /* starting point of a raster line */
                        /* initialize to centre of range */
int xsignedoffset = -128, ysignedoffset = -128; /* signed offset */
int xstepsize = 1, ystepsize = 1; /* spacing between image pts */
                        /* initialize to 1 bit */
int xincfactor = 1;      /* scanning left or right? */
int yincfactor = -1;     /* scanning up or down? */
int pconst = 0, iconst = 0; /* integral and proportional gains */
int setpt = 0;           /* feedback stuff */
int sum = 0, oldsum = 0; /* running integral variables */
                        /* initialize to Z = 0 */
int zin = 0, rin = 0;    /* sampled inputs */
int zout = 0;            /* piezo voltage */
int feedcounter = 0;     /* count how many times feedback has */

```

```

/* been updated at this point */
int feedvalue = 32; /* scan rate = 1/(2*256*feedvalue*SCItimerperiod) */
int xfrac;
int xinc = 0, yinc = 0;
int zclamp = -131071; /* voltage limit to prevent depoling the piezo */
/* initialize to 50 V */
int zarray[512]; /* z(z) curve data array */
int lift = 0; /* height in LiftMode */
int mode = 0; /* Mode: 0 = normal mode; otherwise, LiftMode */
int zzero = 0; /* z voltage between the operating point and */
/* the zero deflection point */
int endpoint; /* pointer to end of real data in zarray */
int zeropoint; /* pointer to zero deflection in zarray */
int delay = 0; /* delay after starting LiftMode */

main()
{
    /*
    movep #$0001,PBC ; Configure Port B for Host Interface
    movep #$00,HCR ; disable Host Receive Data interrupts,
    ; disable Host Transmit Data interrupts,
    ; disable Host Command interrupts
    ; (clear HRIE, HTIE, HCIE)
    ; clear Host Flags

    ; Stepper motor initialization
    movep #$000000,PCC ; Configure Port C for general purpose I/O
    movep #$07,PCDDR ; set pins PC0, PC1, PC2 as outputs,
    ; set the rest as inputs
    movep #$00,PCD ; write 0's to Port C output pins

    movep #$0404,BCR ; Initialize Bus Control Register
    ; External X memory: 0 wait states
    ; External Y memory: 4 wait states
    ; External P memory: 0 wait states
    ; External I/O memory: 4 wait states
    ; The external I/O memory is
    ; Y:$FFC0 - Y:$FFFF and should be
    ; accessed with a MOVEP instruction
    ; if the external I/O wait states
    ; are desired.

    movep #$4400,IPR ; Initialize Interrupt Priority Register
    ; (ie. set Interrupt Priority Levels
    ; for each interrupting device.)
    ; SCI: IPL 0
    ; SSI: disabled
    ; HI: IPL 0
    ; ~IRQB: level triggered, disabled
    ; ~IRQA: level triggered, disabled

    movec #$0000,SR ; clear the Condition Code Register,
    ; unmask all interrupts,
    ; disable scaling and tracing,
    ; clear the Loop Flag

```

```

    movep #$04,HCR           ; enable Host Command interrupts (set HCIE)

    movep #$0000,SCR         ; disable SCI Idle Line interrupts,
                             ; disable SCI Receive Data interrupts,
                             ; disable SCI Transmit Data interrupts,
                             ; disable SCI Timer interrupts
                             ; (clear ILIE, RIE, TIE, TMIE)

    movep #CD,SCCR           ; set SCI timer interrupt rate
    %/
    /* Initialize Image Data to Zero */
    for (i=511; i>=0; i--) darray[i] = 0;

    /* Initialize xfrac to 1/feedvalue */
    /*
    move #0.03125,x0
    move x0,+xfrac
    */

    /* Retract Z before initializing the other DACs */
    /*
    move #>$07ffff,x0
    move x0,CDA3
    move x0,+zout
    move x0,+sum
    rep #24           ; wait for parallel-to-serial conversion
    nop
    */

    /* Initialize the DAC's to the centre of their */
    /* range to start */
    /*
    clr      a
    move a1,CDA1           ; initialize X to 0V
    move a1,CDA2           ; initialize Y to 0V
    move a1,CDA4           ; relax Out 4 DAC
    move a1,CDA5           ; relax Out 5 DAC
    move a1,CDA6           ; relax Out 6 DAC
    move #>$0000ff,x0
    move x0,FDPF           ; Initialize Digitally Prog. Filter to 51.2kHz
    */

    for (;;);           /* infinite loop */
} /* end program */

/***** Interrupt Driven Routines are Grouped in the Following Files *****/

#include "feedback.c"
#include "interrupt.c"
#include "utils.c"
#include "scansize.c"
#include "vector.c"

/* Notes on inline assembly code */

/* According to the Motorola DSP56KCC Cross Compiler User's Manual,

```

DSP56KCC always reserves from 3 to 6 address registers. Programmers wanting to use the address registers should begin with r3 and r7, which are always free.

Labels starting with F, Y, and _, and labels of the form GLx, Lx, LTx, and SIx where x is a number should be avoided as they are automatically generated by the compiler.

The DSP56000 simulator SIM56000.EXE is an excellent way to figure out what the DSP is actually doing.

The interface is very similar to ADS56000.EXE.

Simply LOAD your .LOD file, CHANGE PC to \$40, and STEP through your program, watching the registers change.

You can also compile the program with dsp56kcc -y to generate assembly code with the C code included as comments.

If you're really desperate, the .LOD file is human-readable. */


```

/* FEEDBACK.C */

feedback()
{
/* subroutine to update feedback */
/* Called by SCI Timer interrupt vector at p:$001c */
/*
; READ DATA
move    CAD5,x0      ; start conversion on channels 5 and 6
move    CAD3,x0      ; start conversion on channels 3 and 4
move    CAD1,x0      ; start conversion on channels 1 and 2
rep     #ADC_CONV
nop
move    CAD6,y0      ; read the digitized value from PSPD
move    CAD4,y1      ; read channel 4
move    CAD2,x0      ; read channel 2
move    y0,+zin       ; store y0 in zin
move    y1,+rin       ; store y1 in rin

; DECIDE WHAT TO DO
move    +mode,a       ; load mode into a1
tst     a             Y:Yxincfactor,b ; parallel move xincfactor into b1
jge     NORMAL_MODE   ; if mode >= 0 then do feedback loop
tst     b
jge     LIFT_MODE      ; if xincfactor >= 0 then do lift mode
]NORMAL_MODE

; CALCULATE THE ERROR
move    +zin,x0       ; load zin into x0
move    +setpt,a       ; load setpt into a1
sub     x0,a          Y:Yiconst,x1     ; error = setpt - zin
; parallel move iconst into x1

; UPDATE THE RUNNING SUM (INTEGRAL)
move    a1,y0         ; load error into y0
move    +sum,b         ; load sum into b1
move    b1,+oldsum     ; save old sum in case of saturation
macr    y0,x1,b Y:Ypconst,y1         ; b = sum + iconst*error
; parallel move pconst into y1

; CALCULATE THE P I FEEDBACK (Z VOLTAGE)
macr    y0,y1,b b1,Y:Ysum ; b = (sum + iconst*error) + pconst*error
; parallel move b1 into sum

jmp     CHECK_RANGE
]LIFT_MODE

; FEED ADC2 THROUGH TO DAC6
move    x0,CDA6        ; assume feedthrough is in x0
move    +delay,a       ; send feedthrough to channel 6
tst     a             Y:Yxfrac,x1     ; load xfrac into x1 (positive fraction)
jle     DONE_DELAY     ; if delay <= 0, do lift mode
move    #>$000001,x0
sub     x0,a           ; else decrement delay and jump out
move    a1,+delay
jmp     EXIT_FEEDBACK
]DONE_DELAY

; CALCULATE THE Z VOLTAGE

```

```

; assume xfrac is in x1
move    +feedcounter,x0 ; load feedcounter into x0
; (nonnegative integer < 1/xfrac)
mpy     x0,x1,b          ; b1 = SIGNEDINT(feedcounter*xfrac) = 0
; b0 = UNSIGNEDFRAC(feedcounter*xfrac)
lsl     b                &darray,a
; Convert b0 to a signed fraction
; parallel move a pointer to darray[0] into a
asr     b                Y:Ydatacounter,x0
; Since we know the product is nonnegative,
; we can ignore the negative case.
; parallel move datacounter into x0
add     x0,a            b0,y0 ; load a pointer to darray[datacounter] into a
; parallel move
; SFRAC(feedcounter*xfrac) into y0
move    a1,r3           ; move pointer into address register
move    x0,b            ; Due to pipelining,
; if an address register is changed with a
; MOVE instruction, the new contents are not
; available for use as a pointer until
; the second following instruction.
; (DSP56000 User's Manual)
; move datacounter into b
move    Y:(r3)-,a       ; move darray[datacounter] into a
; postdecrement the address register
tst     b                (r3)- ; parallel move to decrement the address register
jgt     NOT_FIRST       ; if datacounter > 0 then
; don't increment the address register
move    (r3)+           ; increment the address register
move    (r3)+           ; increment the address register
]NOT_FIRST
move    Y:(r3),x0       ; move the previous point into x0
sub     x0,a            x0,b  ; a = darray[datacounter] - previous point
; parallel move the previous point into b
move    a1,y1           ; move difference into y1
macr    y0,y1,b Y:Ylift,x0
; b = previous point +
; difference*SFRAC(feedcounter*xfrac)
; parallel move lift into x0
add     x0,b            Y:Yzzero,x0
; b = b + lift
; parallel move zzzero into x0
add     x0,b            ; b = b + zzzero
]CHECK_RANGE
; CLIP Z TO (ZCLAMP, 2^19-1)
move    #>$07ffff,x0   ; load "-200V" into x0
; assume zout is in b1
cmp     x0,b            ; is zout >= $07ffff?
jlt     CLAMP_Z         ; if not skip next 2 instructions
move    x0,b            ; else zout=$07ffff; ie. clip it
move    +oldsum,x1      ; if saturated don't integrate any more
move    x1,+sum         ; keep the old integral value
jmp     OUTPUT_Z        ; don't check for negative saturation
]CLAMP_Z
move    +zclamp,x0      ; load zclamp into x0
; assume zout is in b1

```

```

cmp      x0,b          ; is zout =< zclamp ?
jgt      OUTPUT_Z      ; if not skip next instruction
move     x0,b          ; else zout=zclamp; ie. clamp it
move     +oldsum,x1     ; same as in positive saturation since we
move     x1,+sum        ; want to stop integrating if either happens
]OUTPUT_Z

move     b1,+zout       ; store b1 in zout
move     b1,CDA3        ; send it to Z DAC

; CALCULATING X RASTER SIGNAL
move     +xinc,x0       ; load xinc into x0 (nonnegative integer)
move     +xfrac,x1      ; load xfrac into x1 (positive fraction)
mpy      x0,x1,b Y:Yxstepsize,y0 ; b1 = SIGNEDINT(xinc*xfrac)
; b0 = UNSIGNEDFRAC(xinc*xfrac)
; parallel move xstepsize into y0
lsl      b              ; Convert b0 to a signed fraction;
asr      b              ; Since we know the product is nonnegative,
; we don't need to worry about the
; negative case.
move     b1,y1          ; move SINT(xinc*xfrac) into y1
mpy      y0,y1,a
asr      a              b0,y1
; required for signed int * signed int,
; see "Fractional and Integer Arithmetic
; Using the DSP56000 Family of General-
; Purpose Digital Signal Processors"
; a10 = SINT(xstepsize*SINT(xinc*xfrac))
; parallel move SFRAC(xinc*xfrac) into y1
move     a0,a           ; result is less than 24 bits; move it into a1
macr     y0,y1,a Y:Yxsignedoffset,x1
; a1 = ROUND(xstepsize*SFRAC(xinc*xfrac) +
; xstepsize*SINT(xinc*xfrac))
; parallel move xsignedoffset into x1
add      x1,a          Y:Yxincfactor,b
; a1=xsignedoffset+xstepsize*xinc*xfrac
; parallel move xincfactor into b1
move     a1,CDA1        ; output the new X value

; INCREMENTING X RASTER COUNTER
add      x0,b          Y:Yfeedcounter,a ; add x0 with a1 ...Result in b1
; parallel move feedcounter into a1
move     b1,+xinc       ; xinc = xinc + xincfactor

; INCREMENTING THE # OF FEEDBACKS COUNTER
move     #>$000001,y0   ; load 1 into y0
add      y0,a          Y:Yfeedvalue,y1 ; increment feedcounter
; parallel move feedvalue into y1

; JUMP OUT UNTIL ENOUGH FEEDBACKS ARE DONE
cmp      y1,a          a1,Y:Yfeedcounter ; compare feedcounter with feedvalue
; parallel move a1 into feedcounter
jlt      EXIT_FEEDBACK ; jump out if feedcounter < feedvalue

; ENOUGH FEEDBACKS...RESET FEEDBACK COUNTER
clr      a             Y:Yxincfactor,b ; clear a
; parallel move xincfactor into b

```

```

tst      b      a1,Y:Yfeedcounter
                                ; parallel move a1 into feedcounter
jle      DOUBLE_INC      ; if xincfactor <= 0 then don't save any data

                                ; SAVE Z VOLTAGE IN DATA ARRAY
move     +mode,a      ; load mode into a1
tst      a
jlt      INC_ZOUT      ; if mode < 0 then don't save zout
move     &darray,a      ; load darray-pointer into a1
move     +datacounter,x1 ; load datacounter into x1
add      x1,a      Y:Yzout,x0      ; offset darray-pointer by datacounter
                                ; parallel move zout into x0
move     a1,r3      ; move the datapointer in r3
nop      ; Due to pipelining,
                                ; if an address register is changed with a
                                ; MOVE instruction, the new contents are not
                                ; available for use as a pointer until
                                ; the second following instruction.
                                ; (DSP56000 User's Manual)
move     x0,Y:(r3)      ; darray[datacounter] = zout
] INC_ZOUT

                                ; INCREMENT DATA COUNTER
move     +xincfactor,x0 ; load xincfactor (i.e scan direction)
move     +datacounter,a ; load datacounter into a1
add      x0,a      Y:Ymode,b ; add xincfactor to datacounter, result in a1
                                ; parallel move mode into b1

tst      b      a1,Y:Ydatacounter
                                ; parallel move a1 into datacounter
jgt      INC_RIN      ; if mode > 0 then don't save rin
                                ; SAVE R VOLTAGE IN DATA ARRAY (RESISTANCE FOR SRM)
move     &darray,a      ; load darray-pointer into a1
move     +datacounter,x1 ; load datacounter into x1
add      x1,a      Y:Yrin,x0 ; offset darray-pointer by datacounter
                                ; parallel move rin into x0
move     a1,r3      ; move the datapointer into r3
nop      ; Due to pipelining,
                                ; if an address register is changed with a
                                ; MOVE instruction, the new contents are not
                                ; available for use as a pointer until
                                ; the second following instruction.
                                ; (DSP56000 User's Manual)
move     x0,Y:(r3)      ; darray[datacounter] = rin
] INC_RIN

                                ; INCREMENT DATA COUNTER
move     +xincfactor,x0 ; load xincfactor (i.e scan direction)
move     +datacounter,a ; load datacounter into a1
add      x0,a      ; add xincfactor to datacounter, result in a1
move     a1,+datacounter ; datacounter = datacounter + xincfactor
jmp      CHECK_ROW_START
] DOUBLE_INC
move     +xincfactor,x0 ; load xincfactor into x0
move     +datacounter,a ; load datacounter into a1
add      x0,a
add      x0,a
move     a1,+datacounter ; datacounter = datacounter + 2*xincfactor

```

```

]CHECK_ROW_START
    ; CHECK TO SEE IF WE HAVE REACHED THE BEGINNING OF A ROW
    ; IF YES CHANGE X SCAN DIRECTION, RESTART DATA COUNTER
    ; AND EXIT FEED BACK ROUTINE
    move    +datacounter,a    ; load datacounter into a1
    tst     a                 ; calculate datacounter - 0
    jgt     CHECK_ROW_END    ; if datacounter > 0 then exit this check

    move    +xincfactor,a     ; load xincfactor into a1
    neg     a                 ; change scandirection
    move    a1,+xincfactor    ; xincfactor = -xincfactor

    clr     b                 Y:Ymode,a ; clear b
                                ; parallel move mode into a1

    tst     a                 b1,Y:Ydatacounter
                                ; parallel move b1 into datacounter
    jle     CHANGE_MODE      ; if mode <= 0 then skip next section

    ; LIFT MODE: OBTAIN Z(Z) CURVE
    ; Tip withdraw
    move    #>$f80000,a       ; load "+200V" into a1
    move    #>$001000,x0      ; load increment into x0
    move    +zout,x1          ; load zout into x1
    move    #>$07ffff,y1      ; load "-200V" into y1
    move    &zarray,r7        ; load pointer to zarray into r7
]LIFT_WITHDRAW
    cmp     x1,a
    jlt     LIFT_READ_UP      ; if a < zout then don't output a
    move    a1,CDA3           ; send a to the DAC
    rep     #LIFT_STEP
    nop

]LIFT_READ_UP
    move    CAD5,y0           ; start conversion on channels 5 and 6
    rep     #ADC_CONV
    nop
    move    CAD6,b            ; load the deflection sensor output into b1
    move    b1,Y:(r7)+        ; store b1 in zarray and postincrement pointer
    add     x0,a              ; a = a + x0
    cmp     y1,a
    jlt     LIFT_WITHDRAW     ; if a < "-200V" then loop again

    ; Tip approach
    move    #>$07f000,a       ; load "-200V" (approx.) into a1
    move    #>$f80000,y1      ; load "+200V" into y1
    move    +setpt,x1         ; load setpt into x1
]LIFT_APPROACH
    cmp     x1,b
    jgt     LIFT_READ_DOWN    ; if deflection > set point then don't output a
    move    a1,CDA3           ; send a to the DAC
    rep     #LIFT_STEP
    nop
    move    r7,+endpoint      ; store endpoint
]LIFT_READ_DOWN
    move    CAD5,y0           ; start conversion on channels 5 and 6
    rep     #ADC_CONV

```

```

nop
move    CAD6,b          ; load deflection sensor output into b1
move    b1,Y:(r7)+      ; store b1 in zarray and postincrement pointer
sub     x0,a            ; a = a - x0
cmp     y1,a
jge     LIFT_APPROACH   ; if a >= "+200V" then loop again

; LIFTMODE:
; CALCULATE THE Z VOLTAGE OFFSET BETWEEN THE OPERATING POINT
; AND THE ZERO DEFLECTION POINT
; Find the deflection sensor output which corresponds to
; zero deflection.
move    +endpoint,r7    ; load endpoint into r7
move    #>48,x1         ; load fuzziness into x1
]FIND_FLATLINE
move    Y:(r7)-,a        ; load first point into a1,
                        ; postdecrement pointer
move    Y:(r7)-,y0       ; load second point into y0,
                        ; postdecrement pointer
sub     y0,a    Y:(r7)+,b ; a = a - y0
                        ; parallel move third point into b,
                        ; postincrement pointer
sub     y0,b            ; b = b - y0
cmpm    x1,a
jge     FIND_FLATLINE   ; if abs(a-x1) > approx. zero then loop again
cmpm    x1,b
jge     FIND_FLATLINE   ; if abs(b-x1) > approx. zero then loop again
move    (r7)+           ; increment r7
move    Y:(r7),x1       ; load zero deflection level into x1

; Find where the deflection sensor output is less than or equal to
; the zero deflection level.
move    +endpoint,r7    ; load endpoint into r7
nop                                ; Due to pipelining,
                                ; if an address register is changed with a
                                ; MOVE instruction, the new contents are not
                                ; available for use as a pointer until
                                ; the second following instruction.
                                ; (DSP56000 User's Manual)
]FIND_ZERO_LEVEL
move    Y:(r7)-,a        ; load point into a, postdecrement pointer
cmp     x1,a
jgt     FIND_ZERO_LEVEL ; if the point is greater than the endpoint
                        ; value, then loop again
move    (r7)+           ; increment r7
move    r7,+zeropoint    ; store r7 in zeropoint

; Find the z voltage of the point where
; the deflection sensor output returns to its initial value.
move    +endpoint,b      ; load endpoint into b1
move    #>511,x1         ; load 511 into x1
move    &zarray,a        ; load pointer to zarray into a1
add     x1,a            ; offset pointer by 511
move    a1,r7           ; load pointer to zarray[511] into r7
move    #>$f80000,a      ; load "+200V" into a1
]FIND_END

```

```

    add    x0,a    (r7)-    ; a = a + x0
                                ; parallel move to decrement r7
    move   r7,x1                ; transfer r7 to x1
    cmp    x1,b                ; 
    jne    FIND_END            ; if b != x1 then loop again
    move   a1,y0                ; transfer z voltage to y0

    ; Find the z voltage of the point where
    ; the deflection sensor output is less than or equal to
    ; the zero deflection level.
    move   +zeropoint,b        ; load zeropoint into b1
]FIND_ZERO
    add    x0,a    (r7)-    ; a = a + x0
                                ; parallel move to decrement r7
    move   r7,x1                ; transfer r7 to x1
    cmp    x1,b                ; 
    jlt    FIND_ZERO          ; if b < x1 then loop again

    sub    y0,a                ; a = zero_deflection - zout
    move   a1,+zzero           ; store zzzero

                                ; MOVE THE TIP TO THE STARTING POSITION FOR LIFTMODE
    ; Tip withdrawal
    move   #>$f80000,a        ; load "+200V" into a1
    move   +zout,x1            ; load zout into x1
    move   #>$07ffff,y1       ; load "-200V" into y1
]LIFT_WITHDRAW2
    cmp    x1,a                ; 
    jlt    LIFT_READ_UP2      ; if a < zout then don't output a
    move   a1,CDA3            ; send a to the DAC
    rep    #LIFT_STEP
    nop
]LIFT_READ_UP2
    add    x0,a                ; a = a + x0
    cmp    y1,a                ; 
    jlt    LIFT_WITHDRAW2     ; if a < "-200V" then loop again

    ; Tip approach
    move   +zzero,a            ; load zzzero into a
    add    x1,a    Y:Ylift,y0    ; a = zout + zzzero
                                ; parallel move lift into y0
    add    y0,a                ; a = zout + zzzero + lift
    move   a1,y1                ; transfer a1 to y1
    move   #>$07f000,a        ; load "-200V" (approx.) into a1
]LIFT_APPROACH2
    move   a1,CDA3            ; send a1 to the DAC
    rep    #LIFT_STEP
    nop
    sub    x0,a                ; a = a - x0
    cmp    y1,a                ; 
    jgt    LIFT_APPROACH2     ; if a > starting position then loop again

    ; Set up a counter to
    ; feed through a number of samples from ADC2 to DAC6
    ; before starting LiftMode,
    ; so that the lock-in amplifier has time to settle.

```

```

        move    #>FEED_DELAY,a
        move    a1,+delay      ; store delay

]CHANGE_MODE
        move    +mode,a        ; load mode into a1
        neg     a
        move    a1,+mode       ; mode = -mode

        jmp     EXIT_FEEDBACK  ; EXIT FEEDBACK ROUTINE

]CHECK_ROW_END
        ; CHECK SEE IF WE HAVE REACHED THE END OF A ROW
        ; IF NO EXIT FEEDBACK ROUTINE, IF YES CHANGE X SCAN DIRECTION
        move    #>$0001ff,x0    ; load 511 into x0 ( 2 images of 256 pixels)
        move    +datacounter,a  ; load datacounter into a1
        cmp     x0,a           ; calculate datacounter - 511
        jle     EXIT_FEEDBACK  ; if datacounter <= 511 then EXIT FEEDBACK

        move    +xincfactor,a   ; load xincfactor into a1
        neg     a              Y:Ymode,b ; change scandirection
                                ; parallel move mode into b1
        move    a1,+xincfactor  ; xincfactor = -xincfactor

                                ; assume $0001ff is in x0
        tst     b              x0,Y:Ydatacounter
                                ; parallel move x0 into datacounter
        jgt     EXIT_FEEDBACK  ; if mode > 0 then EXIT
        jeq     CHECK_Y
        move    +zout,x0        ; load zout into x0
        move    x0,+sum         ; store x0 in sum

]CHECK_Y
        ; CHECK IF WE HAVE REACHED THE TOP OR BOTTOM OF THE IMAGE
        ; AND CHANGE Y SCAN DIRECTION
        move    +yinc,a         ; load yinc into a1
        move    #>$0000ff,x0    ; load 255 into x0
        cmp     x0,a           Y:Ystepsize,x1 ; calculate yinc-255
                                ; parallel move ystepsize into x1
        jeq     CHANGE_Y       ; if yinc=255 CHANGE Y SCAN DIRECTION
                                ; assume yinc is in a1
        tst     a              Y:Yyinc,x0      ; calculate a1-0
                                ; parallel move yinc into x0
        jne     OUTPUT_Y       ; if yinc <> 0 EXIT CHECK

]CHANGE_Y
        move    +yincfactor,a   ; load yincfactor (Y SCAN DIRECTION)
        neg     a              Y:Yyinc,x0      ; change y scan direction
                                ; parallel move yinc into x0
        move    a1,+yincfactor  ; yincfactor = -yincfactor

]OUTPUT_Y
        ; CALCULATING Y RASTER SIGNAL
                                ; assume yinc is in x0
                                ; assume ystepsize is in x1
        mpy     x0,x1,a         ; calculate yinc*ystepsize
        asr     a              Y:Yysignedoffset,y0
                                ; Required for signed int * signed int.
                                ; See *Fractional and Integer Arithmetic

```



```

                                ; Using the DSP56000 Family of General-
                                ; Purpose Digital Signal Processors".
                                ; parallel move ysignedoffset into y0
move    a0,a                    ; result is less than 24 bits; move it into a1
add     y0,a    Y:Yincfactor,b  ; calculate ysignedoffset+yinc*ystepsize
                                ; parallel move yincfactor into b1
move    a1,CDA2                ; output the new Y value

                                ; INCREMENTING Y RASTER COUNTER (ROW/LINE COUNTER)
add     x0,b                    ; add x0 (= yinc) to a1 ...Result in a1
move    b1,+yinc                ; yinc = yinc + yincfactor

                                ; START DATA TRANSFER TO PC
                                ; SET DATA POINTER TO BEGINNING, ENABLE HOST INTERRUPTS OF
                                ; DSP AND VICE-VERSA...EVERYTIME HTX LOW BYTE IS READ DSP
                                ; IS INTERRUPTED AND INT. SERVICE ROUTINE UPDATES THE DATA
movep   #$0000,SCR              ; disable SCI Timer interrupts (clear TMIE)
                                ; until all the data is transferred
move    &darray,r3              ; point data to start
movep   #$06,HCR                ; enable Host Transmit Data interrupts
                                ; (set HTIE)
movep   b1,HTX                  ; send yinc to start transfer and interrupt PC

]EXIT_FEEDBACK
    rti                          ; return from subroutine
%/
}
/* ----- */

setpoint()
{
/* read in new setpoint from PC */
/* called by CV to p:$003c */
/%
    movep   HRX,x0
    move    x0,+setpt            ; store set point in setpt
    rti                          ; return from subroutine
%/
}
/* ----- */

changebias()
{
/* changes bias voltage on channel 4 */
/* called by CV to p:$0034 */
/%
    movep   HRX,x0              ; read in bias voltage
    move    x0,CDA4             ; set bias voltage
    rti                          ; return from subroutine
%/
}
/* ----- */

changeifedback()
{
/* Reads the I-feedback value from PC & updates it */

```

```
/* called by CV to p:$0038          */
/%
    movep    HRX,x0
    move     x0,+iconst      ; store integral gain in iconst
    rti      ; return from subroutine
%/
}
/* ----- */

changefeedback()
{
/* Reads the P-feedback value from PC & updates it */
/* called by CV to p:$0036          */
/%
    movep    HRX,x0
    move     x0,+pconst      ; store proportional gain in pconst
    rti      ; return from subroutine
%/
}
```

```

/* INTERRUPT.C */

enabletimerint()
{
/* Change Serial Communications Interface (SCI) Control Register, */
/* enabling SCI Timer interrupts. ie scanning */
/* called by Command Vector to p:$000c */
/*
    movep    #$2000,SCR        ; set TMIE
    rti                      ; return from subroutine
*/
}
/* ----- */

enableint()
{
/* change Host Command Register on DSP, */
/* enabling Host Transmit Data interrupts */
/* called by Command Vector to p:$002a */
/*
    movep    #$06,HCR          ; enable Host Transmit Data interrupts
                                ; (set HTIE)
    rti                      ; return from subroutine
*/
}
/* ----- */

disableint()
{
/* change Host Command Register on DSP, */
/* disabling Host Transmit Data interrupts */
/* called by a Command Vector to p:$002c */
/*
    movep    #$04,HCR          ; disable Host Transmit Data interrupts
                                ; (clear HTIE)
    rti                      ; return from subroutine
*/
}
/* ----- */

startstop()
{
/* start/stop the AFM - if stop is TRUE, stop it else start it */
/* called by Command Vector to p:$002e */
/*
    movep    HRX,a            ; read flag to start/stop AFM
    tst      a                ; if stop..stop scanning
    jeq      START            ; else start..
    movep    #$0000,SCR        ; disable SCI Timer interrupts (clear TMIE)
    jmp      EXIT_STARTSTOP    ; end stop
]START
    movep    #$2000,SCR        ; enable SCI Timer interrupts (set TMIE)
]EXIT_STARTSTOP
    rti                      ; return from subroutine
*/
}

```

```

/* SCANSIZE.C */

changeyrange()
{
/* Read in the new ystepsize and calculate the new ysignedoffset. */
/* Called by Command Vector to p:$0032 */
/%
    movep HRX,y0                ; Read in new ystepsize
    move y0,+ystepsize

    move #>$07ffff,y1
    move y1,CDA3                ; retract probe
    move y1,+zout
    move y1,+sum
    move #>128,y1                ; load number of steps/2 in 1 image
    mpy -y0,y1,b                ; calculate -range/2
    asr b                        ; required for signed int * signed int,
                                ; see "Fractional and Integer Arithmetic
                                ; Using the DSP56000 Family of General-
                                ; Purpose Digital Signal Processors"

    move b0,b                    ; result in b0 move it up to b1
    move +yoffset,y1             ; load yoffset into y1
    add y1,b                     ; calculate corrected offset for
    move b1,+ysignedoffset       ; signed output (b1 -> b1+y1)

    ; CHANGE THE Y VOLTAGE
    move +yinc,y1                ; load yinc into y1
    mpy y0,y1,a                  ; a = yinc * ystepsize
    asr a    b1,x0               ; Required for signed int * signed int.
                                ; See "Fractional and Integer Arithmetic
                                ; Using the DSP56000 Family of General-
                                ; Purpose Digital Signal Processors".
                                ; parallel move ysignedoffset into x0
    move a0,a                    ; move result into a1
    add x0,a                     ; a = yinc * ystepsize + ysignedoffset
    move a1,CDA2                 ; send the new y voltage to the DAC

    ; SET THE Z CLAMP VALUE
                                ; find the minimum y voltage
    move +yoffset,a              ; load yoffset into a
    tst a
    jgt YRYOPOS                  ; if yoffset > 0
    neg b                        ; b = -ysignedoffset
    jmp YRYOEND
}YRYOPOS
    move #>256,y1                ; load 256 into y1
    mpy y0,y1,a                  ; a = 256 * ystepsize
    asr a                        ; required for signed int * signed int
                                ; see "Fractional and Integer Arithmetic
                                ; Using the DSP56000 Family of General-
                                ; Purpose Digital Signal Processors"
    move a0,y0                   ; move result into y0
    add y0,b                     ; b = ysignedoffset + 256 * ystepsize
}YRYOEND
                                ; find the minimum x voltage
    move +xoffset,a              ; load xoffset into a

```

```

        tst a
        jgt YRXOPOS          ; if xoffset > 0
        move +xsignedoffset,a ; load xsignedoffset into a
        neg a                ; a = -xsignedoffset
        jmp YRXOEND
]YRXOPOS
        move +xstepsize,x0    ; load xstepsize into x0
        mpy x0,y1,a           ; a = 256 * xstepsize
        asr a                 ; required for signed int * signed int
                                ; see "Fractional and Integer Arithmetic
                                ; Using the DSP56000 Family of General-
                                ; Purpose Digital Signal Processors"
        move a0,a             ; move result up into a1
        move +xsignedoffset,x0 ; load xsignedoffset into x0
        add x0,a              ; a = 256 * xstepsize + xsignedoffset
]YRXOEND
                                ; determine which minimum voltage is
                                ; more negative and set zclamp to
                                ; the minimum voltage + 200V
        move #>$f80000,x1     ; load 200V into x1
        move a1,x0            ; load a1 into x0
        cmp x0,b              ; calculate b-x0
        jlt L200              ; if xlimit > ylimit
        add x1,b              ; b = ysignedoffset + 256 * ystepsize + 200V
        jmp L201
]L200
        add x1,a              ; a = 256 * xstepsize + xsignedoffset + 200V
        move a,b              ; transfer a to b
]L201
                                ; clip the clamp value to 200V
        cmp x1,b              ; calculate b-x1
        jge EXIT_YRANGE       ; if proposed zclamp < 200V
        move x1,b             ; set zclamp to 200V
]EXIT_YRANGE
        move b1,+zclamp       ; store zclamp

        rti                  ; return from subroutine
%/
}
/* ----- */

changeyoff()
{
/* Changes the Y offset to a value dictated by PC */
/* Called by Command Vector to p:$0028 */
/%
        movep HRX,y1          ; read in Y offset value
        move y1,+yoffset

        move #>$07ffff,x0
        move x0,CDA3          ; retract probe
        move x0,+zout
        move x0,+sum
        move +ystepsize,y0    ; load ystepsize into y0
        move #>128,x0         ; load number of steps/2 in 1 image
        mpy -y0,x0,b          ; calculate -range/2

```

```

asr b                                ; required for signed int * signed int
                                    ; See "Fractional and Integer Arithmetic
                                    ; Using the DSP56000 Family of General-
                                    ; Purpose Digital Signal Processors"
move b0,b                            ; result in b0, move it to b1
add y1,b                             ; calculate corrected offset for
move b1,+ysignedoffset               ; signed output ( b1->b1+y1 )

; CHANGE THE Y VOLTAGE
move +yinc,y1                       ; load yinc into y1
mpy y0,y1,a                         ; a = yinc * ystepsize
asr a    b1,x0                      ; Required for signed int * signed int.
                                    ; See "Fractional and Integer Arithmetic
                                    ; Using the DSP56000 Family of General-
                                    ; Purpose Digital Signal Processors".
                                    ; parallel move ysignedoffset into x0
move a0,a                            ; move result into a1
add x0,a                             ; a = yinc * ystepsize + ysignedoffset
move a1,CDA2                        ; send the new y voltage to the DAC

; SET THE Z CLAMP VALUE
                                    ; find the minimum y voltage
move +yoffset,a                     ; load yoffset into a
tst a
jgt YOYOPos                         ; if yoffset > 0
neg b                               ; b = -ysignedoffset
jmp YOYOEND

]YOYOPos
move #>256,y1                       ; load 256 into y1
mpy y0,y1,a                         ; a = 256 * ystepsize
asr a                               ; required for signed int * signed int
                                    ; see "Fractional and Integer Arithmetic
                                    ; Using the DSP56000 Family of General-
                                    ; Purpose Digital Signal Processors"
move a0,y0                          ; move result into y0
add y0,b                            ; b = ysignedoffset + 256 * ystepsize
]YOYOEND

                                    ; find the minimum x voltage
move +xoffset,a                     ; load xoffset into a
tst a
jgt YOXOPos                         ; if xoffset > 0
move +xsignedoffset,a               ; load xsignedoffset into a
neg a                               ; a = -xsignedoffset
jmp YOXOEND

]YOXOPos
move +xstepsize,x0                  ; load xstepsize into x0
mpy x0,y1,a                         ; a = 256 * xstepsize
asr a                               ; required for signed int * signed int
                                    ; see "Fractional and Integer Arithmetic
                                    ; Using the DSP56000 Family of General-
                                    ; Purpose Digital Signal Processors"
move a0,a                            ; move result up into a1
move +xsignedoffset,x0              ; load xsignedoffset into x0
add x0,a                            ; a = 256 * xstepsize + xsignedoffset
]YOXOEND

; determine which minimum voltage is

```

```

                                ; more negative and set zclamp to
                                ; the minimum voltage + 200V
                                ; load 200V into x1
                                ; load a1 into x0
                                ; calculate b-x0
                                ; if xlimit > ylimit
                                ; b = ysignedoffset + 256 * ystepsize + 200V
                                ; clip the clamp value to 200V
                                ; calculate b-x1
                                ; if proposed zclamp < 200V
                                ; set zclamp to 200V
                                ; store zclamp
                                ; return from subroutine

                                %/
                                }
                                /* ----- */

changexrange()
{
/* Updates xrange variable & calculates xfrac=xrange/xincrange */
/* effectively changing X step size. */
/* Called by Command Vector to p:$0030 */
/%
    movep HRX,x0                ; Read in new xstepsize
    move x0,+xstepsize

    move #>$07ffff,x1
    move x1,CDA3                ; retract probe
    move x1,+zout
    move x1,+sum
    move #>128,x1                ; load number of steps/2 in 1 image
    mpy -x0,x1,a                ; calculate -range/2
    asr a                        ; required for signed int * signed int
                                ; see "Fractional and Integer Arithmetic
                                ; Using the DSP56000 Family of General-
                                ; Purpose Digital Signal Processors"
    move a0,a                    ; result in a0 move it up to a1
    move +xoffset,x1            ; load xoffset into x1
    add x1,a                     ; calculate corrected offset for
    move a1,+xsignedoffset      ; signed output (a1 -> a1+x1)

    ; CHANGE THE X VOLTAGE
    move +xinc,x1                ; load xinc into x1
    move +xfrac,y0              ; load xfrac into y0
    mpyr x1,y0,b                 ; b = xinc * xfrac
    move b1,y0                   ; move integer part into y0
    mpy x0,y0,b                  ; b = xinc * xfrac * xstepsize
    asr b    a1,y0               ; Required for signed int * signed int.
                                ; See "Fractional and Integer Arithmetic

```

```

; Using the DSP56000 Family of General-
; Purpose Digital Signal Processors".
; parallel move xsignedoffset into y0
move b0,b          ; move result into b1
add y0,b           ; b = xinc * xfrac * xstepsize + xsignedoffset
move b1,CDA1       ; send the new x voltage to the DAC

; SET THE Z CLAMP VALUE

; find the minimum x voltage
move +xoffset,b    ; load xoffset into b
tst b
jgt XRXOPOS        ; if xoffset > 0
neg a              ; a = -xsignedoffset
jmp XRXOEND

]XRXOPOS
move #>256,x1      ; load 256 into x1
mpy x0,x1,b        ; b = 256 * xstepsize
asr b              ; required for signed int * signed int
; see "Fractional and Integer Arithmetic
; Using the DSP56000 Family of General-
; Purpose Digital Signal Processors"
move b0,x0         ; move result into x0
add x0,a           ; a = xsignedoffset + 256 * xstepsize
]XRXOEND

; find the minimum y voltage
move +yoffset,b    ; load yoffset into b
tst b
jgt XRYOPOS        ; if yoffset > 0
move +ysignedoffset,b ; load ysignedoffset into b
neg b              ; b = -ysignedoffset
jmp XRYOEND

]XRYOPOS
move +ystepsize,y0 ; load ystepsize into y0
mpy y0,x1,b        ; b = 256 * ystepsize
asr b              ; required for signed int * signed int
; see "Fractional and Integer Arithmetic
; Using the DSP56000 Family of General-
; Purpose Digital Signal Processors"
move b0,b          ; move result up into b1
move +ysignedoffset,y0 ; load ysignedoffset into y0
add y0,b           ; b = 256 * ystepsize + ysignedoffset
]XRYOEND

; determine which minimum voltage is
; more negative and set zclamp to
; the minimum voltage + 200V
move #>$f80000,y1  ; load 200V into y1
move b1,y0         ; load b1 into y0
cmp y0,a           ; calculate a-y0
jlt L206           ; if ylimit > xlimit
add y1,a           ; a = xsignedoffset + 256 * xstepsize + 200V
jmp L207

]L206
add y1,b           ; b = 256 * ystepsize + ysignedoffset + 200V
move b,a           ; transfer b to a

]L207

; clip the clamp value to 200V

```



```

        cmp y1,a                ; calculate a-y1
        jge EXIT_XRANGE         ; if proposed zclamp < 200V
        move y1,a                ; set zclamp to 200V
EXIT_XRANGE
        move a1,+zclamp          ; store zclamp

        rti                      ; return from subroutine
    */
}
/* ----- */

changexoff()
{
/* Changes the X offset to a value dictated by PC */
/* Called by Command Vector to p:$0026 */
/%
        movep HRX,x1             ; read in X offset value
        move x1,+xoffset

        move #>$07ffff,y0
        move y0,CDA3             ; retract probe
        move y0,+zout
        move y0,+sum
        move +xstepsize,x0        ; load xstepsize into a
        move #>128,y0             ; load number of steps/2 in 1 image
        mpy -x0,y0,a             ; calculate -range/2
        asr a                    ; Required for signed int * signed int.
                                   ; See "Fractional and Integer Arithmetic
                                   ; Using the DSP56000 Family of General-
                                   ; Purpose Digital Signal Processors"

        move a0,a
        add x1,a                 ; calculate corrected offset for
        move a1,+xsignedoffset    ; signed output

        ; CHANGE THE X VOLTAGE
        move +xinc,x1             ; load xinc into x1
        move +xfrac,y0           ; load xfrac into y0
        mpyr x1,y0,b             ; b = xinc * xfrac
        move b1,y0               ; move integer part into y0
        mpy x0,y0,b              ; b = xinc * xfrac * xstepsize
        asr b a1,y0              ; Required for signed int * signed int.
                                   ; See "Fractional and Integer Arithmetic
                                   ; Using the DSP56000 Family of General-
                                   ; Purpose Digital Signal Processors".
                                   ; parallel move xsignedoffset into y0
        move b0,b                ; move result into b1
        add y0,b                 ; b = xinc * xfrac * xstepsize + xsignedoffset
        move b1,CDA1             ; send the new x voltage to the DAC

        ; SET THE Z CLAMP VALUE
                                   ; find the minimum x voltage
        move #>256,x1             ; load 256 into x1
        mpy x0,x1,b              ; b = 256 * xstepsize
        asr b                    ; required for signed int * signed int
                                   ; see "Fractional and Integer Arithmetic
                                   ; Using the DSP56000 Family of General-

```

```

                                ; Purpose Digital Signal Processors"
move b0,x0                    ; move result into x0
add x0,a                      ; a = xsignedoffset + 256 * xstepsize

                                ; find the minimum y voltage
move +ystepsize,y0           ; load ystepsize into y0
mpy y0,x1,b                   ; b = 256 * ystepsize
asr b                         ; required for signed int * signed int
                                ; see "Fractional and Integer Arithmetic
                                ; Using the DSP56000 Family of General-
                                ; Purpose Digital Signal Processors"
move b0,b                     ; move result up into b1
move +ysignedoffset,y0       ; load ysignedoffset into y0
add y0,b                      ; b = 256 * ystepsize + ysignedoffset

                                ; determine which minimum voltage is
                                ; more negative and set zclamp to
                                ; the minimum voltage + 200V
move #>$f80000,y1            ; load 200V into y1
move b1,y0                    ; load b1 into y0
cmp y0,a                      ; calculate a-y0
jlt L209                      ; if ylimit > xlimit
add y1,a                      ; a = xsignedoffset + 256 * xstepsize + 200V
jmp L210

L209                          ; b = 256 * ystepsize + ysignedoffset + 200V
add y1,b                      ; transfer b to a
move b,a

L210                          ; clip the clamp value to 200V
                                ; calculate a-y1
cmp y1,a                      ; if proposed zclamp < 200V
jge EXIT_XOFF                 ; set zclamp to 200V
move y1,a

EXIT_XOFF                     ; store zclamp
move a1,+zclamp

                                ; return from subroutine
rti

%/
}
/* ----- */

changefeedvalue()
{
/* Calculates xfrac=1/feedvalue and updates feedvalue, which in turn */
/* defines the scan rate since #feedbacks/row=feedvalue*256 */
/* Called by Command Vector to p:$003a */
/%

movep HRX,x0                  ; read in new feedvalue
move x0,+feedvalue

move #>$07ffff,a
move a1,CDA3                  ; retract probe
move a1,+zout
move a1,+sum
clr a                          ; clear a
move a1,+xinc                  ; reset xinc i.e start scanning row
move a1,+datacounter           ; reset datacounter i.e. no data

```

```
    move a1,+feedcounter    ; reset feedcounter => we're starting
    move #>$000001,a        ; set a1 to 1
    move a1,+xincfactor      ; scan direction is "+" increasing X

    andi #$fe,ccr           ; prepare for long division
                             ; by clearing the C bit of the CCR
    rep  #$18               ; repeat 24 times for 24 bit precision
    div  x0,a               ; calculate 1/feedvalue
    move a0,+xfrac          ; save result in xfrac

    rti                    ; return from subroutine
%/
}
```

```

/* UTILS.C */

stepper()
{
/* Single step the stepper motor in either direction */
/* Called by Command Vector to p:$0018 */
/%
    movep HRX,a          ; read in step direction
    move #>MOTOR_STEP,x0 ; load number of clock cycles/2 to wait
                        ; between state changes of step bit

    tst a                ; is it going down?
    jeq STEP0            ; else step up..

    move #>$3,a           ; set PC2,PC1,PC0 to 011, 0=/en,1=step,1=dir
    movep a1,PCD          ; send step command to stepper controller
    rep x0                ; wait x0 times 2 clock cycles
    nop

    move #>$1,a           ; set PC2,PC1,PC0 to 001, 0=/en,0=step,1=dir
    movep a1,PCD          ; send step command to stepper controller
    rep x0                ; wait x0 times 2 clock cycles
    nop

    jmp EXIT_STEPPER      ; stepping done; exit
]STEP0
    move #>$2,a           ; set PC2,PC1,PC0 to 010, 0=/en,1=step,0=dir
    movep a1,PCD          ; send step command to stepper controller
    rep x0                ; wait x0 times 2 clock cycles
    nop

    clr a                ; set PC2,PC1,PC0 to 000, 0=/en,0=step,0=dir
    movep a1,PCD          ; send step command to stepper controller
    rep x0                ; wait x0 times 2 clock cycles
    nop

]EXIT_STEPPER
    move #>$4,x0          ; load motor-off bit-mask
    or x0,a              ; or mask with step out to preserve direction
    movep a1,PCD          ; sent stepout variable to stepper motor
    nop                  ; turning it off
    nop

    rti                  ; return from subroutine
%/
}
/* ----- */

sendmeter()
{
/* Send photodiode meter value to PC */
/* value is 18 bits sign extended to 24...2s complement */
/* Called by Command Vector to p:$001a */
/%
    move CAD5,x0          ; start conversion Chnls 5&6
    rep #ADC_CONV
    nop

```

```

        move CAD6,x0          ; read the value in from the ADC
        movep x0,HTX          ; write value to host data register

        rti                   ; return from subroutine
    %/
}
/* ----- */

changepcutoff()
{
/* Send 8 bit number to digitally programmable filter */
/* the cutoff=(value of binary number/256)*51.2KHz */
/* Called by Command Vector to p:$0024 */
/*
        movep    HRX,a          ; read cutoff number from PC
        move     a1,FDPF        ; send it to xilinx

        rti                   ; return from subroutine
    %/
}
/* ----- */

changelift()
{
/* Change the lift height */
/* Called by Command Vector to p:$0020 */
/*
        movep    HRX,a          ; read lift height from PC
        move     a1,+lift       ; store lift
        move     a1,+mode       ; if lift = 0 then mode = 0
                                ; else mode > 0

        rti                   ; return from subroutine
    %/
}
/* ----- */

setz()
{
/* Set Z voltage */
/* Called by Command Vector to p:$0014 */
/*
        movep    HRX,a          ; read Z voltage from PC
        move     a1,CDA3        ; send it to the piezo
        move     a1,+zout       ; store zout
        move     a1,+sum        ; store sum

        rti                   ; return from subroutine
    %/
}
/* ----- */

zzcurve()
{
/* Acquire a z(z) curve and send it to the PC */
/* Called by Command Vector to p:$0016 */

```

```

/%
    ; Initialize variables
    ; Piezo begins pulling away from the tip
    move    #>$f80000,a      ; load +200V into a1
    move    #>$001000,x0     ; load increment into x0
    move    +zout,x1         ; load zout into x1
    move    #>$07ffff,y1     ; load -200V into y1
    move    &zarray,r7       ; load pointer to zarray into r7
]WITHDRAW
    cmp     x1,a
    jlt     READ_UP          ; if a < zout then don't output a
    move    a1,CDA3          ; send a to the DAC
]READ_UP
    rep     #ZZ_STEP
    nop
    move    CAD5,y0          ; start conversion on channels 5 and 6
    rep     #ADC_CONV
    nop
    move    CAD6,b           ; load the deflection sensor output into b1
    move    b1,CDA6
    move    b1,Y:(r7)+       ; store b1 in zarray and postincrement pointer
    add     x0,a             ; a = a + x0
    cmp     y1,a
    jlt     WITHDRAW        ; if a < -200V then loop again

    ; Initialize variables
    ; Piezo begins moving toward the tip
    move    #>$07f000,a      ; load -200V (approx.) into a1
    move    #>$f80000,y1     ; load +200V into y1
    move    +setpt,x1        ; load setpt into x1
]APPROACH
    cmp     x1,b
    jgt     READ_DOWN        ; if deflection > set point then don't output a
    move    a1,CDA3
]READ_DOWN
    rep     #ZZ_STEP
    nop
    move    CAD5,y0          ; start conversion on channels 5 and 6
    rep     #ADC_CONV
    nop
    move    CAD6,b           ; load deflection sensor output into b1
    move    b1,CDA6
    move    b1,Y:(r7)+       ; store b1 in zarray and postincrement pointer
    sub     x0,a             ; a = a - x0
    cmp     y1,a
    jge     APPROACH        ; if a >= +200V then loop again

    move    +zout,x1
    move    x1,CDA3          ; send original zout to piezo

    move    &zarray,r3       ; load pointer to zarray into r3
    movep    #$06,HCR        ; enable Host Transmit Data interrupts
                                ; (set HTIE)
    movep    +zout,HTX       ; start transfer and interrupt PC
                                ; by sending zout

```

```
        rti                ; return from subroutine
%/  
}
```

```

/* VECTOR.C */

/* Reset vector */
/* After reset, this is where the program counter points */
/* Jump to start of real program */
/%
    section reset
    org    p:$0000          ; where to put this routine
    jmp    $0040            ; jump to start of real program
    nop                      ; it's a 2-word interrupt
    endsec
%/
/* ----- */

/* SSI Receive Data interrupt */
/* May be used as a Host Command interrupt if RIE in the CRB is clear */
/* Call routine to enable SCI Timer interrupts */
/%
    section enabletimerinterrupt
    org    p:$000c          ; where to put this routine
    jsr    Fenabletimerint+4 ; jump to enabletimerint()
                                ; it's a 2-word interrupt
    endsec
%/
/* ----- */

/* SSI Receive Data with Exception Status interrupt */

/* ----- */

/* SSI Transmit Data interrupt */

/* ----- */

/* SSI Transmit Data with Exception Status interrupt */

/* ----- */

/* SCI Receive Data interrupt */
/* May be used as a Host Command interrupt if RIE in the SCR is clear */
/* Call routine to set Z voltage to a value read in from the PC */
/%
    section zinit
    org    p:$0014          ; where to put this routine
    jsr    Fsetz+4          ; jump to setz()
                                ; it's a 2-word interrupt
    endsec
%/
/* ----- */

/* SCI Receive Data with Exception Status interrupt */
/* May be used as a Host Command interrupt if RIE in the SCR is clear */
/* Call routine to acquire a z(z) curve */
/%
    section zz

```


[illegible]

```

%/
/* ----- */

/* Host Transmit Data interrupt */
/* Enabled by HTIE in the HCR */
/* send data from DSP to PC */
/%
    section senddatum
    org    p:$0022        ; where to put this routine
    movep  y:(r3)+,HTX    ; move data at pointer to output
    nop                    ; it's a 2-word interrupt
    endsec

%/
/* ----- */

/* default Host Command interrupt */
/* Enabled by HCIE in the HCR */
/* Call routine to change Dig. Prog. Filter cutoff frequency */
/%
    section filtercutoff
    org    p:$0024        ; where to put this routine
    jsr    Fchangecutoff+4 ; jump to changecutoff()
                                ; it's a 2-word interrupt
    endsec

%/
/* ----- */

/* Host Command interrupt */
/* Enabled by HCIE in the HCR */
/* Call routine to change X scan offset */
/%
    section xoffset
    org    p:$0026        ; where to put this routine
    jsr    Fchangexoff+4  ; jump to changexoff()
                                ; it's a 2-word interrupt
    endsec

%/
/* ----- */

/* Host Command interrupt */
/* Enabled by HCIE in the HCR */
/* Call routine to change Y scan offset */
/%
    section Yoffset
    org    p:$0028        ; where to put this routine
    jsr    FchangeYoff+4  ; jump to changeYoff()
                                ; it's a 2-word interrupt
    endsec

%/
/* ----- */

/* Host Command interrupt */
/* Enabled by HCIE in the HCR */
/* Call routine to enable Host Transmit Data interrupts */
/%
    section enable

```

```

        org    p:$002A          ; where to put this routine
        jsr    Fenableint+4      ; jump to enableint()
                                   ; it's a 2-word interrupt
    endsec

%/
/* ----- */

/* Host Command interrupt */
/* Enabled by HCIE in the HCR */
/* Call routine to disable Host Transmit Data interrupts */
/%

    section disable
        org    p:$002C          ; where to put this routine
        jsr    Fdisableint+4    ; jump to disableint()
                                   ; it's a 2-word interrupt
    endsec

%/
/* ----- */

/* Host Command interrupt */
/* Enabled by HCIE in the HCR */
/* Call routine to toggle SCI timer enable */
/%

    section togglefeedback
        org    p:$002E          ; where to put this routine
        jsr    Fstartstop+4     ; jump to startstop()
                                   ; it's a 2-word interrupt
    endsec

%/
/* ----- */

/* Host Command interrupt */
/* Enabled by HCIE in the HCR */
/* Call routine to change X range */
/%

    section xrange
        org    p:$0030          ; where to put this routine
        jsr    Fchangexrange+4  ; jump to changexrange()
                                   ; it's a 2-word interrupt
    endsec

%/
/* ----- */

/* Host Command interrupt */
/* Enabled by HCIE in the HCR */
/* Call routine to change Y range */
/%

    section yrange
        org    p:$0032          ; where to put this routine
        jsr    Fchangeyrange+4  ; jump to changeyrange()
                                   ; it's a 2-word interrupt
    endsec

%/
/* ----- */

/* Host Command interrupt */

```

[illegible]

```
%/  
/* ----- */  
  
/* Host Command interrupt */  
  
/* ----- END CODE -----*/
```

Appendix B

SPM image file format

Each image acquired by the SPM control system is recorded in two files: a binary file containing the raw data, and a text file which describes the data. Both files are given the same name but different suffixes.

The binary data file is composed of a stream of 65 536 16-bit signed integers in Intel byte order, with values along the x -axis stored first. This file is given a .EXP suffix.

The ASCII parameter file is composed of several lines of information regarding its associated data file. Each line is terminated with a carriage return and line feed, which signal the end of a line in DOS. If this file is transferred to another platform, the end-of-line characters should be translated to the native format to maintain its readability by humans. The information normally contained in this file is described in Table B.1. Only the first 10 lines are read by the data analysis program. This file is given a .INF suffix.

Table B.1. Revised INF file format.

Example	Description
Untitled	Image description
0.00	STM bias potential in volts
0.15	Set point in volts
39990.3	<i>x</i> dimension in nanometres
39990.3	<i>y</i> dimension in nanometres
256	Number of samples in <i>x</i>
256	Number of samples in <i>y</i>
23276	Maximum value in EXP
-23278	Minimum value in EXP
2486.29	<i>z</i> dimension in nanometres or volts, as appropriate
800.0	Lift height in nanometres
1997:08:15	Date image acquired
12:24:40	Time image acquired