Pattern Dependencies in the Plasma Etching of Polysilicon

by

Timothy Joseph Dalton

B.E. Chemical Engineering
Stevens Institute of Technology, 1988

M.S. Chemical Engineering Practice
Massachusetts Institute of Technology, 1990

Submitted to the Department of Chemical Engineering
in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy
in
Chemical Engineering

at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
September, 1994

© Massachusetts Institute of Technology, 1994. All Rights Reserved.

Signature of Author____
Department of Chemical Engineering

Certified by____
Professor Herbert H. Sawin, Thesis Advisor
Departments of Chemical Engineering and
Electrical Engineering and Computer Science

Accepted by____
Professor Robert E. Cohen
Chairman, Committee for Graduate Students

ARCHIVES
SEP 2 1994

Pattern Dependencies in the Plasma Etching of Polysilicon

by

Timothy Joseph Dalton

Submitted to the Department of Chemical Engineering on
June 14, 1994 in Partial Fulfillment of the Requirements for
the Degree of Doctor of Philosophy in Chemical Engineering

Abstract

The radio frequency magnetically-enhanced plasma etching of polycrystalline silicon (polysilicon) in an HBr/Cl_2 chemistry was investigated using 100-mm wafers on an Applied Materials Precision 5000 (AME-5000) plasma etcher. Statistically designed experiments were used to construct a neural network model for the responses of the etching process.

The formation of microtrenches during the etching of a polysilicon gate over a sub-100 Å oxide layer was studied. A model was developed to account for the observed phenomena. The specular reflection of ions from photoresist and polysilicon sidewall features during etching can account for the formation of the observed microtrenches. The model qualitatively explained the dependence of microtrench formation on feature geometry.

A new plasma diagnostic technique (Full Wafer Interferometry, FWI) was developed to measure etch rates and uniformity *in situ* during plasma etching. Plasma optical emission (OE) was used as a light source; the entire wafer was imaged through an optical bandpass filter onto a CCD camera during the etching process. The interference of the OE reflected off of the wafer and the OE transmitted through a polysilicon film and then reflected back was used to measure the etch rates as a function of position during etching. The etch rates measured using this technique compared quite well with those measured by other *in situ* techniques (laser interferometry with a photodiode detector) but had the advantage of yielding spatially resolved results. This technique has potential application in real time process control and in process development.

Reduced temperature etching of polysilicon in pure Cl_2 was studied. Heat transfer in the AME-5000 was examined and the heat transfer coefficient for wafer-to-cathode heat transfer was determined. Wafer and cathode pedestal temperatures were measured *in situ* using Fluoroptic® probes. The wafer temperature showed two different responses: a rapid initial response due to wafer heating that was well described by an exponential increase with a time constant on the order of 10 seconds; and a long, slow linear increase due to the heating of the pedestal that was bolted to the cooled cathode base. Ion current

densities and plasma source efficiencies were estimated from temperature measurements in an Ar plasma and ranged from 0.1 to 1.5 mA cm^{-2} and 2 to 20%, respectively, as a function of input conditions. The activation energy for polysilicon etching was less than 1 kcal/mol, indicating a largely ion-enhanced chemistry.

Wafer surface and plasma potentials were measured during plasma etching using a Langmuir probe constructed from RF chokes operating near their self-resonance frequency. In the absence of a magnetic field, no variations were observed in the DC wafer surface potential. However, in the presence of a magnetic field, significant variations (on the order of 20 V) were observed in the direction perpendicular to the magnetic field. The variations correlated with magnetic field direction, and were also observed in the bulk plasma potential. The variations were caused by the drift of electrons and can be detrimental to device performance. Such variations were minimized by increasing pressure, decreasing power, and decreasing magnetic field strength.

Aspect ratio dependent etching (ARDE) was studied using FWI. Grating patterns of different size were monitored at the wafer center with the FWI apparatus in a high resolution mode to measure feature etch rates as a function of the feature size *in situ*. Both "RIE Lag" (smaller features etch faster than large features) and "Inverse RIE Lag" were observed. Variations as small as 1-2 percent of the total etching rate could be observed. Such variations are difficult if not impossible to measure using conventional *ex situ* methods such as SEM photographs of a partial etch.

ARDE depended upon a number of input parameters (residence time (feed gas flowrate), pressure, magnetic field strength, temperature and feed gas composition) as well as on uncontrolled parameters (chamber seasoning/history). Models were developed to explain the behavior of ARDE on the input parameters.

Thesis Supervisor: Professor Herbert H. Sawin
MIT Professor of Chemical Engineering and Electrical Engineering and Computer Science

Dedication

To my Parents,
For Helping Me Become Who I am Today
and for Their Sacrifices for Me.

I Only Wish My Father Could Have Seen This Day...

Acknowledgments

"What A Long Strange Trip Its Been!" But, it has finally come to an end. It seems like it has been years, and it has. But all of the long hours and hard work have finally paid off. After almost 23 straight years of being a student, it is now time to leave school and become a useful member of society (yeah, right ;-)

First, my thanks to Heidi, for putting up with me, and all of my weird habits, long hours, and missed nights and weekends while working at MIT. It was tough, but the hardest part is over. Now, we can try and begin a "Normal" life. To you, all of my love and gratitude (and first few paychecks ;-)

And also first, to my parents, who worked hard to see that I could get ahead. Years of struggling, so that I could go to a good high school, and then to college, and finally grad school. They helped me reach where I am today. For providing me with resources as a child, encouraging questions, going to the library, sometimes doing without so that I might have something. This day seemed like it would never get here, but it finally has.

To Prof. Herb Sawin, my advisor during my time here at MIT, for all of his help and guidance in defining this project and pushing me on with it. His insight and ideas helped make this work a success. And also for his taking care of all of the really hard work of getting the funds and the best equipment to make this project possible. And I'd also like to thank my committee members: Prof. Karen Gleason and Prof. Klavs Jensen of this department, and Dr. Ed Gleason of Lincoln Labs.

To Gerry Gladu, for all of the weekends of fun time. Whether it be hiking in the White Mountains, driving across the country, brewing some beers, taping a show, or just drinking way too much beer, here, in Germany, Amsterdam, England, or wherever.

To my friends and community on "The Wheel", for providing discussion, humor, and a view of the outside.

To Jerry Garcia and the Grateful Dead, for providing the hours upon hours of music entertainment for me during my work, and for the shows that provided the needed distractions to keep my sane while working here. All that I ask now is that you play St. Stephen and The Eleven one more time!

To MIT, for offering the environment where technonerds can flourish and pursue their ideas and goals with some of the best facilities available.

To the people in METL who made everyday life at MIT pass a little faster. Those who came before me to show the way to freedom - Dave, Igor, Joanne, Linda, Sub, Bill, J.P., and Evangelos. And to those who are here now struggling to reach the outside - Gil, Vivek, John, and Gib, and to those who may be struggling to reach the outside - Gavin.

To the support staff members: Arline, Nancy, Carol, Peter, and Glori.

To Steve Swan and Dan Corliss at Digital Equipment Corporation for their part in the work on microtrench formation (Chapter 4).

And to John Arnold, the coauthor on the microtrench work (Chapter 4).

To Bill Conner, for his part in the FWI of Chapter 5.

To some of the good people in the Boston Wort Processors homebrew club and Brew Free of Die, for our monthly (or more frequent) meetings to relax, enjoy some good beers and B.S. This calls for a few Foghorns to celebrate!

And to those who provided funding for my work here - SEMATECH, The National Science Foundation, and Tau Beta Pi.

Table of Contents

Abstract . . . 2
Dedication . . . 4
Acknowledgments . . . 5
List of Figures . . . 10
List of Tables . . . 22

1. Introduction . . . 24
1.1 Microelectronics Fabrication . . . 24
1.2 Etching . . . 27
1.3 Plasmas . . . 27
1.4 Plasma Etching of Polysilicon . . . 32
1.4.1 Fluorine and Chlorine . . . 35
1.4.2 Bromine and Iodine . . . 37
1.4.3 Temperature Effects . . . 39
1.4.4 Kinetic Studies . . . 40
1.4.5 Mask Material . . . 40
1.5 Plasma/Surface Interactions . . . 40
1.5.1 Ion Energy and Angular Distributions (IED/IAD) . . . 41
1.5.2 Neutral Flux . . . 45
1.5.3 Surface Reactions : Ion-Neutral Synergism . . . 47
1.5.4 Neutral Transport . . . 50
1.6 Pattern Dependent Etching . . . 54
1.6.1 Undercutting Variations With Line Spacing . . . 57
1.6.2 Aspect Ratio Dependent Etching . . . 59
1.6.3 Microtrenching, Notching . . . 59
1.6.4 Other Microscopic Pattern Dependencies . . . 60
1.6.5 Full Wafer Uniformity . . . 62
1.7 Goals of This Research . . . 62

2. Experimental Apparatus and Diagnostics . . . 64
2.1 Installation of AME-5000 . . . 66
2.1.1 Gas Handling - Etchants, Pneumatic Air, and Purge N_2 . . . 66
2.1.2 Other Utilities . . . 70
2.2 Plasma Diagnostics . . . 71
2.2.1 Laser Interferometry . . . 73
2.2.2 Optical Emission Spectroscopy . . . 77
2.2.3 RF Power Monitor . . . 86
2.2.4 Wafer and Cathode Temperature . . . 93
2.2.5 Full Wafer Interferometry . . . 94
2.3 Data Acquisition . . . 96
2.3.1 Data Acquisition Hardware . . . 96
2.3.2 Data Acquisition Software . . . 98
2.4 Optical Emission Characterization of Plasmas . . . 98

2.4.1 Long Range, High Resolution Plasma Scan - HR640. 98
2.4.2 Xinix and DARSS Scans 99
2.4.3 Actinometry 115
2.4.4 End Point 127
2.5 Etching Test Mask 127

3. Characterization of Polysilicon Etching in HBr/Cl_2 133
3.1 Process Response Analysis 133
3.1.1 Design of Experiments (DOE) 133
3.1.2 Response Surface Methodology (RSM). 134
3.1.3 Neural Networks (NNAPER) 135
3.2 Experimental Design 137
3.2.1 Experimental Layout 137
3.2.2 Measurements 140
3.3 Neural Network Construction and Solution 140
3.4 Plasma Electrical Parameter Characterization 146
3.4.1 Electrical Analog Model 146
3.4.2 Results and Discussion 150
3.5 Etching Rate and Uniformity 161
3.6 Cl and Br Concentrations 168
3.7 Conclusions. 179

4. Microtrench Formation During Plasma Etching 181
4.1 Introduction 181
4.2 Experimental Procedures and Results 184
4.3 Model Formulation 187
4.3.1 Possible Causes of Microtrenches 187
4.3.2 Profile Modeling 189
4.3.3 Ion Flux Calculation 189
4.3.4 2-D vs. 3-D Calculations 192
4.3.5 Ion Angular Distribution 194
4.3.6 Ion Specular Reflection Probability. 196
4.4 Model Results 201
4.5 Comparison to Experiment 207
4.6 Conclusions. 210

5. Full Wafer Interferometry 213
5.1 Introduction 213
5.1.1 Physics of Interferometry 214
5.1.2 Interferometry Techniques 217
5.2 Experimental Method 220
5.2.1 CCD SYSTEM 220
5.2.2 CCD Interferometry. 222
5.2.3 Plasma Etcher and Magnetic Field Synchronization 226
5.2.4 Software for Data Analysis 227

5.2.5 Etching Rate Calculation 227
5.3 Results 230
5.3.1 Measurement Noise and Comparison to Other Techniques. . . . 230
5.3.2 Analysis Area Effects 237
5.3.3 Observation of Pattern Dependent Etching 240
5.3.4 Interferometric Images 240
5.3.5 Uniformity Across a Die 244
5.3.6 Uniformity Across a Wafer 244
5.3.7 Effect of Film Thickness 254
5.4 Determining End-Point Time and Uniformity 257
5.4.1 Introduction to End-Point Determination Methods 257
5.4.2 End-Point Experiments 262
5.4.3 Comparison of End-Point Methods 272
5.5 Conclusions. 276

6. Reduced Temperature Polysilicon Etching 282
6.1 Introduction 282
6.2 Reactor Modifications 283
6.2.1 Temperature Control 285
6.2.2 Temperature Measurement 287
6.3 Thermal Analysis of AME-5000 291
6.3.1 Radiative Heat Transfer. 292
6.3.2 Convective Heat Transfer to Feed Gas 294
6.3.3 Convective Heat Transfer To Pedestal - Conduction Across a
Static Pressurized Gap 294
6.3.4 Heat Transfer Coefficient Determination 297
6.3.5 Wafer Deflection 303
6.3.6 Radial Temperature Variations 306
6.3.7 Wafer Temperature During Plasma Etching 308
6.3.8 Wafer Power Deposition and Ion Current. 313
6.4 Reduced Temperature Polysilicon Etching in Cl_2 321
6.4.1 Etch Rates. 321
6.4.2 Etching Uniformity and Profile 329
6.5 Conclusions. 332

7. Spatially-Resolved Plasma and Wafer-Surface Potential Measurements 334
7.1 Introduction 334
7.2 Experimental Setup 340
7.2.1 Etcher 340
7.2.2 Langmuir Probe 340
7.3 Results 345
7.3.1 Surface Potentials 345
7.3.2 Plasma Potentials 347
7.3.3 Process Variables 356
7.4 Conclusions. 363

8. Aspect Ratio Dependent Etching 367
8.1 Introduction to ARDE 367
8.2 Method of Approach 368
8.2.1 CCD Interferometry 368
8.2.2 Test Mask 369
8.3 Effect of Process Variables 374
8.3.1 Residence Time (Total Flow Rate) 374
8.3.2 Pressure 375
8.3.3 Magnetic Field Strength 378
8.3.4 Temperature 381
8.3.5 Feed Gas Composition 384
8.3.6 Chamber History/Seasoning 388
8.4 Plasma Characterization 390
8.4.1 Plasma Optical Emission Spectroscopy 390
8.4.2 Plasma Chemistry 398
8.4.3 CSTR Model for Plasma Behavior 401
8.5 Discussion 408
8.5.1 Residence Time and Aspect Ratio Dependent Etching 408
8.5.2 Temperature and Aspect Ratio Dependent Etching 410
8.5.3 Shadowing Effects and Aspect Ratio Dependent Etching 415
8.6 Conclusions 419

References 423

Appendices 442

A.1 AME5000 Data Acquisition Code 442

A.2 AME-5000 Data Analysis and Plotting Code 532

A.3 Full Wafer Interferometry Code 602

A.4 Interferometry Model Code 671

A.5 CSTR Model Code 676

A.6 Other Equipment Drawings and Descriptions 681

List of Figures

Figure 1-1. Outline of Major Steps in Subtractive Processing 26

Figure 1-2. Anisotropic and Isotropic Etching Profiles 28

Figure 1-3. Plasma and Its Interactions With a Surface 31

Figure 1-4. Ion Angular Distributions for 10, 50 and 500 mTorr. Distributions are fit to the data of Liu et al. [1990] 44

Figure 1-5. Right and Left Angular View Limits for a Feature, and Their Effect on Ion Flux for Two Different IADs 46

Figure 1-6. Some Typical "Pattern Dependent" Etching Phenomena 56

Figure 1-7. Possible Physical Causes of Some "Pattern Dependent" Etching Phenomena 58

Figure 2-1. Overhead View of AME-5000 With Major Components Indicated. . . . 65

Figure 2-2. Standard Etchant Gas Handling System. 68

Figure 2-3. Plasma Diagnostics Installed on the AME-5000 72

Figure 2-4. Sample Laser Interferometry Signal, and its Second Derivative 75

Figure 2-5. Laser Interferometry Photodetector Amplifier Circuitry 76

Figure 2-6. Plasma Optical Emission Spectrum for HBr/Cl_2 Plasma Taken with Xinix 100-mm Scanning Monochromator (7000-8000 Å). 79

Figure 2-7. Plasma Optical Emission Spectrum for HBr/Cl_2 Plasma Taken with Jarrell Ash Monospec 270-mm Monochromator With Linear Diode Array Showing Components used to Calculated Actual Spectrum: Background Spectrum (Black) and Measured Spectrum (Gray) 82

Figure 2-8. Plasma Optical Emission Spectrum for HBr/Cl_2 Plasma Taken with Jarrell Ash Monospec 270-mm Monochromator With Linear Diode Array . . . 83

Figure 2-9. Plasma Optical Emission Spectrum for HBr/Cl_2 Plasma Taken with Jobin Yvone 640-mm Monochromator (7000-8000 Å) 85

Figure 2-10. RMS values of Plasma RF Voltage and RF Current Measured using a Comdel RPM-1 88

Figure 2-11. Phase Angle Between RF Current and Voltage Waveforms, and Calculated Plasma Power, Measured Using Comdel RPM-1 89

Figure 2-12. Comparison of Comdel RPM Voltage With Tektronix 100x Voltage Probe and Lecroy 9400 Oscilliscope 91

Figure 2-13. Comparison of Comdel RPM Plasma Power and Nominal Plasma Power Measured at the RF Generator for 39 Experiments Over a Wide Range of Experimental Conditions 92

Figure 2-14. Location of Luxtron Temeperature Probes on RF Cathode 95

Figure 2-15. Plasma Optical Emission Spectrum for 100 mTorr HBr (10 SCCM) / Cl_2 (30 SCCM) Plasma taken With Grating "B" on HR640 Monochromator (1750 - 4750 Å) 100

Figure 2-16. Plasma Optical Emission Spectrum for 100 mTorr HBr (10 SCCM) / Cl_2 (30 SCCM) Plasma taken With Grating "A" on HR640 Monochromator (4250-8250 Å). 101

Figure 2-17. Xinix Plasma Optical Emission Spectrum for 100 mTorr HBr (2000-8000 Å). 102

Figure 2-18. DARSS Plasma Optical Emission Spectrum for 100 mTorr HBr (8000-8600 Å). 104

Figure 2-19. DARSS Plasma Optical Emission Spectrum for 100 mTorr HBr (6950-7575 Å). 105

Figure 2-20. Xinix Plasma Optical Emission Spectrum for 100 mTorr Cl_2 (2000-8000 Å). 106

Figure 2-21. DARSS Plasma Optical Emission Spectrum for 100 mTorr Cl_2 (8000-8600 Å). 107

Figure 2-22. DARSS Plasma Optical Emission Spectrum for 100 mTorr Cl_2 (6950-7575 Å). 108

Figure 2-23. Evolution of Br and Cl Signals Over Time in a Cl_2 Plasma following an HBr Plasma 110

Figure 2-24. Xinix Plasma Optical Emission Spectrum for 100 mTorr Ar (2000-8000 Å) 111

Figure 2-25. DARSS Plasma Optical Emission Spectrum for 100 mTorr Ar (8000-8600 Å). 112

Figure 2-26. DARSS Plasma Optical Emission Spectrum for 100 mTorr Ar (6950-7575 Å). 113

Figure 2-27. Xinix Plasma Optical Emission Spectrum for 100 mTorr HBr (10 SCCM)-Cl_2 (30 SCCM), With Assignments in the UV Region (2000-8000 Å) . 114

Figure 2-28. Xinix Plasma Optical Emission Spectrum for 100 mTorr HBr (10 SCCM)-Cl_2 (30 SCCM) Without and With Ar (3 SCCM) (7000-8000 Å). . . . 119

Figure 2-29. DARSS Plasma Optical Emission Spectrum for 100 mTorr HBr (10 SCCM) / Cl_2 (30 SCCM) / Ar (2 SCCM) Showing Lines Used in Actinometry (8000-8600 Å). 120

Figure 2-30. Actinometric Ratio (Q_{Ar} * Br 8375.94 Å / Ar 8115.311 Å) as a Function of Time for Different Ar Flow Rates (2 SCCM = 5%, 3 SCCM = 7.5%) and DARSS Integration Times. 121

Figure 2-31. DARSS Plasma Optical Emission Spectrum for 100 mTorr HBr (10 SCCM) / Cl_2 (30 SCCM) / Ar (2 SCCM) Showing Lines Used in Actinometry (6950-7575 Å). 123

Figure 2-32. Actinometric Ratios Using Peak Heights for 2 Br I and 2 Cl I Lines to Ar I (7503.869 Å) as a Function of Time 125

Figure 2-33. Actinometric Ratios Using Peak Areas for 2 Br I and 2 Cl I Lines to Ar I (7503.869 Å) as a Function of Time (Same Data as Figure 2-32). . . . 126

Figure 2-34. Optical Emission Spectra Near the Br^+ 4704.85 Å Line Used for End-Point Determination. 128

Figure 2-35. Optical Emission Signal for Br^+ 4704.85 Å Line Used for End-Point Determination On All Three of the Monochromators. 129

Figure 2-36. Test Mask Pattern 131

Figure 3-1. Structure of a Three Layer Neural Network. Network consists of Input, Output and Hidden Nodes. After [NNAPER, 1993] 136

Figure 3-2. Evaluation of Neural Network Model Fit to Training and Test Data . . 144

Figure 3-3. Evaluation of Neural Network Model Fit to Polysilicon Positional Etching Rate Data (Uniformity) 145

Figure 3-4. Evaluation of Neural Network Model Fit to Etchant Concentrations from Actinometry and Electrical Parameters from RPM 147

Figure 3-5. Plasma Electrical Analog Model (After [Allen, 1986d]) 148

Figure 3-6. Modeled rms RF Voltage as a Function of Power and Magnetic Field Strength. Pressure = 100 mTorr, Composition = 25% HBr 153

Figure 3-7. Modeled rms RF Current as a Function of Power and Magnetic Field Strength. Pressure = 100 mTorr, Composition = 25% HBr 155

Figure 3-8. Modeled Phase Angle as a Function of Power and Magnetic Field Strength. Pressure = 100 mTorr, Composition = 25% HBr 156

Figure 3-9. Predicted Electron Density as a Function of Power and Magnetic Field Strength. Pressure = 100 mTorr, Composition = 25% HBr 157

Figure 3-10. Comparison of Predicted Electron Density and Ar 7503.869 Å Optical Emission Intensity as a Function of Power 158

Figure 3-11. Comparison of Predicted Electron Density and Ar 7503.869 Å Optical Emission Intensity as a Function of Magnetic Field 160

Figure 3-12. Predicted Electron Density as a Function of Composition (% HBr) and Pressure. P = 250 Watts, B = 75 Gauss. 162

Figure 3-13. Average Etching Rate as a Function of Power and Magnetic Field Strength. Pressure = 100 mTorr, Composition = 25% HBr 165

Figure 3-14. Etching Rate Non-Unifomrity as a Function of Power and Magnetic Field Strength. Pressure = 100 mTorr, Composition = 25% HBr 166

Figure 3-15. Average Etching Rate as a Function of Pressure and Feed Gas Composition (%HBr). Power = 250 W, Magnetic Field = 75 G 169

Figure 3-16. rms RF Voltage as a Function of Pressure and Feed Gas Composition (%HBr). Power = 250 W, Magnetic Field = 75 G 170

Figure 3-17. Etching Rate Non-Unifomrity as a Function of Pressure and Feed Gas Composition (%HBr). Power = 250 W. Magnetic Field = 75 G 171

Figure 3-18. Modeled Bromine Concentration as a Function of Magnetic Field and Power. Pressure = 100 mT. Composition = 25% HBr. 173

Figure 3-19. Modeled Chlorine Concentration as a Function of Magnetic Field and Power. Pressure = 100 mT. Composition = 25% HBr. 174

Figure 3-20. Effect of Magnetic Field on [Br]. Concntrations Normalized to Concentration Without Magnetic Field. (375 Gauss, 25% HBr). . . . 176

Figure 3-21. Modeled Bromine Concentration as a Function of Composition (% HBr) and Pressure. P = 250 Watts, B = 75 Gauss 177

Figure 3-22. Modeled Chlorine Concentration as a Function of Composition (% HBr) and Pressure. P = 250 Watts, B = 75 Gauss 178

Figure 4-1. SEM Photograph of a Typical Line-Space Pattern, Exhibiting Microtrenches 183

Figure 4-2. Cross Section of the Film Structure at Polysilicon Etching End-Point, With Typical Values for Polysilicon and Photoresist Sidewall Angles. . . . 185

Figure 4-3. Definition of View Angles and the Integration of the Ion Flux to the Feature Bottom Between the View Angles 191

Figure 4-4. Nomenclature for Calculation of Reflected Fluxes, Φ_P or Φ_R 193

Figure 4-5. Equivalence of 2-D and 3-D Trajectories for Ions Undergoing Specular Reflection From Sidewall 195

Figure 4-6. 2-D and 3-D Ion Angular Distributions 197

Figure 4-7. Physical Sputtering Yield and Assumed Reflection Probability as a Function of Ion Impingement Angle 199

Figure 4-8. Predicted Ion Flux Profile and Surface Topography For Structure Shown in Figure 1 202

Figure 4-9. Details of Φ_R for Feature Modeled in Figure 4-8. Total Φ_R is the Sum of Ion Fluxes Reflected from Left and Right Sidewall Photoresist 203

Figure 4-10. Response Surface for Maximum Nonuniformity Ratio, Rmax, with Varying Photoresist Angle and Thickness 204

Figure 4-11. Response Surface for Microtrench Offset as a Function of Photoresist and Polysilicon Angles 205

Figure 4-12A. Scanning Electron Micrograph of a Feature Cross Section Showing Asymmetric Microtrenching 208

Figure 4-12B. Modeling of Asymmetric Microtrenches - Predicted Initial Ion Fluxes Along Feature Bottom. 209

Figure 4-12C. Modeling of Asymmetric Microtrenches - Surface Topography After Etching 209

Figure 4-13A. Effect of Line Spacing on Microtrenching - Scanning Electron Micrograph of a 0.35 μm Feature, Showing a Single Microtrench at the Centerline 211

Figure 4-13B. Effect of Line Spacing on Microtrenching - Predicted Initial Ion Flux Profiles Where Film Angles and Thicknesses Are Held Constant . . . 212

Figure 5-1. Film Structure and Variables Used in Interferometry Calculations . . . 215

Figure 5-2. Systems for *in situ* Measurement of Etch Rate During Plasma Etching 218

Figure 5-3. CCD-based Full Wafer Interferometry System 221

Figure 5-4A Plasma Optical Emission and Optical Band Pass Filter Percent Transmission for the 753 nm Filter Used in FWI 224

Figure 5-4B. Plasma Optical Emission and Optical Band Pass Filter Percent Transmission for the 480 nm Filter Used in FWI 225

Figure 5-5. Effect of FFT Order on Calculated Etching Rate 229

Figure 5-6. CCD Signal for a Non-Etching Part of the Chamber 231

Figure 5-7A. Compaision of Interferometry Signals at the Same Location on the Wafer for Optical Emission Interferometry Using the CCD, and for Laser Interferometry Using the CCD and a Photodiode 233

Figure 5-7B. Polysilicon and Photoresist Interferometry Signals at 480 nm, and Modeled Reflectivity of Poly 235

Figure 5-8. Comparison of Etching Rates Determined With FWI and With a Nanospec/AFT 200 236

Figure 5-9. Observed CCD Signal and FFT Magnitude for the Etching of Patterned Polysilicon. 238

Figure 5-10. CCD Signal for the Etching of Unpatterned Polysilicon 239

Figure 5-11. Observation of Pattern Dependent Etching Using FWI 241

Figure 5-12A. Time Sequence of CCD Images During Wafer Etch (11 Seconds) . . . 242

Figure 5-12B. Time Sequence of CCD Images During Wafer Etch (48 Seconds) . . . 242

Figure 5-12C. Time Sequence of CCD Images During Wafer Etch (80 Seconds) . . . 243

Figure 5-12D. Time Sequence of CCD Images During Wafer Etch (97 Seconds) . . . 243

Figure 5-13. Etching Rate Across a Blank Polysilicon Die, Surrounded by Photoresist. Surface and Contour Plots. 244

Figure 5-14. Etching Rates Across an Entire Wafer, Measured in situ. Magnetic Field Strength = 0 Gauss (Surface) 246

Figure 5-15. Etching Rates Across an Entire Wafer, Measured in situ. Magnetic Field Strength = 0 Gauss (Cross Section) 247

Figure 5-16. Etching Rates Across an Entire Wafer, Measured in situ. Magnetic Field Strength = 75 Gauss (Surface) 248

Figure 5-17 Etching Rates Across an Entire Wafer, Measured in situ. Magnetic Field Strength = 75 Gauss (Cross Section) 249

Figure 5-18. Etching Rates Across an Entire Wafer, Measured in situ. Magnetic Field Strength = 75 Gauss, Temperature = -75°C (Surface) 252

Figure 5-19. Etching Rates Across an Entire Wafer, Measured in situ. Magnetic Field Strength = 75 Gauss, Temperature = -75°C (Cross Section) 253

Figure 5-20. Effect of Magnetic Field Strength on Radial Uniformity for Undoped Polysilicon Etching at 375 Watts 255

Figure 5-21. Radial Nonuniformity for Several Runs Over Several Days as Determined by CCD FWI and by Nanospec 256

Figure 5-22. Effect of Underlying Oxide Film Thickness on FWI Signal for Unpatterned Polysilicon at Wafer Center 258

Figure 5-23. Model Results for Effect of Underlying Oxide Film Thickness on Reflectivity 259

Figure 5-24. Diagnostics Installed on AME-5000 263

Figure 5-25. End-Point Time as a Function of Position Across a Wafer, Shown for Two Cross-Sections of the Wafer. B = 0 Gauss 266

Figure 5-26. Measured Plasma Power as a function of Time, With The Matching Network in Both Auto-Tune and "Hold" Modes 268

Figure 5-27. Plasma Optical Emission Signal at 7547 Å as a function of Time . . . 270

Figure 5-28. Measured Difference Between Wafer and Cathode Pedestal Temperatures as a Function of Time both With and Without a Magnetic Field 271

Figure 5-29A. Time to Start of Clearing for Different End-Point Methods. B = 0 G. . 274

Figure 5-29B. Time to End of Clearing for Different End-Point Methods. B = 0 G . . 275

Figure 5-30A. Time to Start of Clearing for Different End-Point Methods. B = 75 G . 277

Figure 5-30B. Time to End of Clearing for Different End-Point Methods. B = 75 G . 278

Figure 6-1. Temperature Range for Anisotropic Plasma Etching of Polysilicon. After Tachi et al. [1991] 284

Figure 6-2. Optical Emission from N_2 First Positive System at 6544.8 Å Showing Air Leak at Low Temperatures 286

Figure 6-3. Temperature Probe Feedthrough for AME-5000 290

Figure 6-4. Effect of Radiative Heat Transfer and Wafer Backside Conduction on Temperature Difference Between Two Luxtron MIW-type Temperature Probes 293

Figure 6-5. Effect of Chamber Pressure on Wafer Cooling After Plasma Etching. 75 SCCM SF6, 4 Torr He 295

Figure 6-6. Calculated Mean Free Path of He (λ_{He}), Wafer Deflection (g_0), and Knudsen Number (N_{Kn}) as a Function of He Backside Pressure 298

Figure 6-7. Typical Temperature Response for HTC Measurement. Room Temperature Wafer Placed on Cold Cathode, then He Turned on . . . 301

Figure 6-8. Wafer Heat Transfer Coefficient as a Function of He Backside Pressure. Circles are "Cold" Calcualtions, and Squares are "Hot" Calculation . . 302

Figure 6-9. Ultimate Wafer Temperature and (T_w - T_p) as a Function of He Pressure. Cl_2, 87 SCCM, 300 mTorr, 475 Watts, 0 Gauss, 20°C Cathode. Also Shown is the Source Efficiency 309

Figure 6-10. Wafer and Pedestal Temperatures and Their Difference as a Function of Time. 500 Watt, 0 Gauss, 100 mTorr, 30 SCCM Cl_2, 10 SCCM HBr, 4 Torr He, 20°C 311

Figure 6-11. Wafer Temperature and Model Fits as a Function of Time 312

Figure 6-12. Ion Currents Estimated From T_w - T_p for a 250 Watt Ar Plasma 316

Figure 6-13. Source Efficiency (Power Deposited in Wafer as a Percent of RPM Measured Plasma Power) Estimated From $T_w - T_p$ for a 250 Watt Ar Plasma 318

Figure 6-14. Ion Currents Estimated From $T_w - T_p$ for a 100 mTorr Ar Plasma. Curve is for 0 Gauss. Points shown for 50, 100, and 150 Gauss at 250 Watts. . 319

Figure 6-15. Source Efficiency (Power Deposited in Wafer as a Percent of RPM Measured Plasma Power) Estimated From $T_w - T_p$ for a 100 mTorr Ar Plasma 320

Figure 6-16. Stability of Ion Currents and Source Efficiency Estimated From $T_w - T_p$ for a 250 Watt, 100 mTorr, 0 Gauss Ar Plasma 322

Figure 6-17. Etch Rate for Blanket and Patterned Undoped Poly and n^{++} (P)-Doped Blanket Poly as a Function of the Measured Wafer Surface Temperature During Etching 324

Figure 6-18. Etch Rate*$T^{0.5}$ for Unpatterned and Patterned Undoped Poly and n^{++} (P)-Doped Unpatterned Poly as a Function of the the Inverse of the Measured Wafer Surface Temperature During Etching 326

Figure 6-19. Comparison of Temperature Dependance of Etching Rates for Phosphorus Doped Polysilicon Using Various Chemistries and Etchers 327

Figure 6-20. Etch Rate Nonuniformity For Undoped Blanket Polysilicon Wafers as a Function of Measured Wafer Temperature During Etching 330

Figure 7-1. Typical MERIE Configuration, Showing Electric and Magnetic Field Directions as Well as the Effective Electron Path, and the Direction of the "E Cross B" Drift of Electrons, Leading the the Production of a Non-Uniform Plasma 337

Figure 7-2. Impedance of RF Chokes as a Function of Frequency for a Number of Components 341

Figure 7-3. Schematic View of Langmuir Probe Feedthrough and Positioning on Etcher 343

Figure 7-4. Test Wafer Design Used to Measure Surface Potential Variations . . . 344

Figure 7-5. Wafer Surface Potential Variations as a Function of Position Without a Magnetic Field 346

Figure 7-6. Wafer Surface Potential Variations as a Function of Position With a 100 Gauss Magnetic Field, Fixed 'Side' 348

Figure 7-7. Plasma Floating Potential as a Function of Axial Position Above Three Metal Pads (Marked on Figure 7-4 as R = 0.0, ±1.7 in.) 349

Figure 7-8. Plasma Floating Potential as a Function of Position over the Metal Pads used to Measure Surface Potential 351

Figure 7-9. Plasma Floating Potential as a Function of Height at Wafer Center for a Cl_2 Discharge With Magnetic Fields of 0 and 75 Gauss 353

Figure 7-10. Plasma Floating Potential Above Center of Wafer in Cl_2 Discharge With 75 Gauss Magnetic Field Rotating at 0.5 and 1.0 Hz 354

Figure 7-11. Plasma Floating Potential as a Function of Position Across Wafer, 0.52 inches Above Wafer. Cl_2 Discharge at 100 G. Note: Field is Fixed 'Front' and Not 'Side' as is Shown With the Wafer Image 355

Figure 7-12. Affect of Pressure and Magnetic Field on Measured Pad Voltage (DC Bias) at Wafer Center in Ar Discharge 357

Figure 7-13. Surface Potential Variations With Process Variables. Wafer Surface Potential Plotted as a Function of Plasma Power and Pressure for a Magnetic Field Strength of 0 Gauss. 360

Figure 7-14. Surface Potential Variations With Process Variables. Wafer Surface Potential Plotted as a Function of Plasma Power and Pressure for a Magnetic Field Strength of 60 Gauss 361

Figure 7-15. Surface Potential Variations With Process Variables. Wafer Surface Potential Plotted as a Function of Plasma Power and Pressure for a Magnetic Field Strength of 120 Gauss 362

Figure 7-16. DC Bias Variations With Process Variables. DC Bias Plotted as a Function of Plasma Power and Pressure for a Magnetic Field Strength of 0 G . . 364

Figure 7-17. DC Bias Variations With Process Variables. DC Bias Plotted as a Function of Plasma Power and Pressure for a Magnetic Field Strength of 60 G . 365

Figure 7-18. DC Bias Variations With Process Variables. DC Bias Plotted as a Function of Plasma Power and Pressure for a Magnetic Field Strength of 120 G 366

Figure 8-1. View of Patterned Wafer Taken With f=16-mm Lens 370

Figure 8-2. View of Wafer Center During Etch, with the Region of the Die used in the ARDE Analysis Indicated, Along with the Location of the Patterns in that Region. 371

Figure 8-3. Test Mask Pattern 373

Figure 8-4. Feature Etching Rate Relative to Unpatterned Polysilcion (Open Field) Etching Rate as a Function of Feature Aspect Ratio for Several Different Flowrates 376

Figure 8-5. Effect of Total Gas Flow Rate on Average Etching Rate. Also shown is the Required Conversion to Support the Measured Etching Rate. 377

Figure 8-6. Effect of Chamber Pressure on Aspect Ratio Dependent Etching . . . 379

Figure 8-7. Effect of Magnetic Field Strength on Aspect Ratio Dependent Etching 380

Figure 8-8. Effect of Cathode Temperature on Aspect Ratio Dependent Etching. . 382

Figure 8-9. Slope of ARDE Curve (Figure 8-8) showing Transition from "RIE Lag" (Negative Slope) to "Inverse RIE Lag" (Positive Slope) 383

Figure 8-10. Etching Rate as a Function of Aspect Ration for Several Cathode Temperatures 385

Figure 8-11. Effect of Gas Phase Composition on ARDE 386

Figure 8-12. Slope of ARDE Curve (Figure 8-11) showing the Effect of Gas Phase Composition on ARDE. Both "RIE Lag" (Negative Slope) and "Inverse RIE Lag" (Positive Slope) Are Observed as Compositon Varies. Etching Temperature was +20°C (Squares). The Diamonds are for etching at -20°C. Point at 75% Cl_2 from Figure 8-9. 387

Figure 8-13. Effect of Temperature And Composition on ARDE 389

Figure 8-14. Affect of Chamber History/Seasoning on ARDE 391

Figure 8-15. Optical Emission Signal for SiCl Radical. Doublet Splitting of the $X^2\Pi_r$ ground state is visible in this $B'^2\Delta$ - $X^2\Pi_r$ Electronic Transition 394

Figure 8-16A. Actinometric Ratios as a Function of Total Flowrate for Cathode Temperatures of +20°C and -25°C. Cl, Cl_2, and Cl^+ OE Ratioed to Ar OE 395

Figure 8-16B. Actinometric Ratios as a Function of Total Flowrate for Cathode Temperatures of +20°C and -25°C. Si, SiCl, and $SiCl_2$ OE Ratioed to Ar OE 396

Figure 8-16C. Actinometric Ratios as a Function of Total Flowrate for Cathode Temperatures of +20°C and -25°C. Br, Br^*, and Br_2^+ OE Ratioed to Ar OE 397

Figure 8-17. Actinometric Ratios for Cl (7414Å and 7547Å) and Br (7005Å and 7349Å) with Simultaneous Laser Interferometry Trace. Cl Rise Corresponds to End-Point. 399

Figure 8-18. Reaction Network For Decomposition of $SiCl_4$ to Form $(SiCl_x)_n$ Polymers. After [Tiller and Sameith, 1990], [Tiller et al., 1984]. 400

Figure 8-19. Calculated Values of the Peclet Number as a Function of Flowrate for an Entire Wafer (l = 5.0 cm) and for a Die (l = 0.5 cm) 403

Figure 8-20. Calculated Values of the Damkohler Number as a Function of Reaction Probability for an Entire Wafer (l = 5.0 cm), for a Die (l = 0.5 cm), and for a Feature. 405

Figure 8-21. CSTR Model Results for Species Concentrations as a Function of Total Flow Rate 407

Figure 8-22. SEM Measured Nominal Line Spacings (µm) as a Function of Total Flow Rate (SCCM) for Nominal Line Spacings of 1.0, 1.5, and 2.0 µm . . . 409

Figure 8-23. SEM Measured Nominal Line Spacings (µm) as a Function of Cathode Temperature (°C) for Nominal Line Spacings of 1.0, 1.5, and 2.0 µm . 413

Figure 8-24. SEM Photographs of 1.0 µm Space Etched at -30°C Showing Sidewall Deposits. A) is Immediately After Etching With the Resist On. B) is After an HF Dip 414

Figure 8-25. Calculated ARDE as a Function of the IAD Standard Deviation. Also Shown is the Data of Figure 8-7. 416

Figure 8-26. ARDE predicted from Neutral Shadowing Model of Gottscho *et al.* [1992] 420

List of Tables

Table 1-1. Halogen Reaction Probabilities 33

Table 1-2. Dopant Etching Rate Effect for Halogens. 34

Table 2-1. Etchant Gases and Flowrate Ranges on AME-5000 67

Table 2-2. AME-5000 Cooling Water Requirements 70

Table 2-3. Diffraction Gratings on Jarrell-Ash Monospec-27 80

Table 2-4. Comparison of Three Different Monochromators 86

Table 2-5. Correlation Coefficients for Different Electrical Parameters 93

Table 2-6. Electronic Transitions Used in Actinometry 124

Table 3-1. Designed Experiment Input Parameters and Ranges 138

Table 3-2. Experimental Conditions for Designed Experiments 139

Table 3-3. Plasma Electrical Property Characterization 152

Table 3-4. Etching Rates, Nonuniformity and Etchant Concentrations 163

Table 5-1. Complex Indices of Refraction for Polysilicon and SiO_2 217

Table 6-1. Maximum Radial Wafer Temperature Differences 307

Table 6-2. Activation Energies for Cl_2 MERIE of Polysilicon 325

Table 6-3. Activation Energies From Other Researchers. 328

Table 7-1. Larmor Radii and Mean Free Paths for Electrons and Ions 338

Table 7-2. Designed Experiment Input Parameters and Ranges 356

Table 7-3. List of Designed Experiments and Outputs 359

Table 8-1. Nominal Process Parameter Values 374

Table 8-2. Aspect Ratio Dependence of Activation Energy 384

Table 8-3. Plasma Species and Optical Emission Lines 393

Table 8-4. IAD Standard Deviation and Dimensionless Sheath Thickness for Ar . 417

Table 8-5. Dimensionless Sheath Thickness for Cl_2 418

Chapter 1

Introduction

This research focused on the pattern dependencies in the plasma etching of polysilicon. The goals were to (1) increase the understanding of pattern dependent etching phenomena, (2) develop *in situ* control monitors for etching to help characterize pattern dependencies, and (3) to develop processes that are pattern independent. In this introduction, I first talk about microelectronics processing (Section 1.1), and then introduce the concept of etching (Section 1.2). Then plasmas are introduced in Section 1.3; plasma etching of polysilicon is then discussed (Section 1.4). Next, plasma/surface interactions are discussed in Section 1.5. Pattern dependent etching phenomena are introduced in Section 1.6. Finally, I present a brief outline of the chapters in this thesis (Section 1.7).

1.1 Microelectronics Fabrication

The semiconductor industry is continually increasing the density of devices on integrated circuit chips to achieve better performance and greater integration. This is accomplished by a reduction in feature size. Presently, devices are fabricated with feature sizes on the order of 0.5 μm while researchers approach 0.1 μm or beyond [Taur, 1994]. The accurate transferral of features from the mask to the underlying film becomes more important as the feature size is reduced, for the successful completion of both that process

step and subsequent processing steps.

The fabrication of microelectronic devices is a complicated multistep process. A typical MOS (metal-oxide-semiconductor) fabrication process may consist of 30 main process steps and more than 100 individual process steps [Sze, 1983]. These steps include ion implantation, oxidation, deposition, lithography, etching, diffusion and annealing. Several etching steps are required to fabricate a working device. Figure 1-1 shows a typical deposition-photolithography-etch cycle. First, as shown in Figure 1-1A, a thin film (e.g., polycrystalline silicon, silicon dioxide, silicon nitride) is deposited onto a substrate (typically, a silicon wafer) using a technique such as chemical vapor deposition (CVD) or sputtering. Next, a photosensitive polymer known as photoresist (or resist) is applied to the surface by spinning on an even coat. The photoresist is exposed to ultraviolet light through a mask which consists of clear and opaque regions. This process is known as photolithography and is shown in Figure 1-1B. The photoresist not exposed to the light is removed from the surface using a chemical solution known as a developer, yielding the surface shown in Figure 1-1C. Such a photoresist is also known as positive photoresist because a positive image of the mask is produced. If the exposed portions of the photoresist remains on the surface after developing, it is known as a negative resist. At this point, the desired pattern has been transferred from the mask to the photoresist. The pattern is then transferred into the underlying film using an etching process (discussed in Section 1.2). This is shown in Figure 1-1D. Finally, the resist mask is removed (stripped) from the thin film to yield the desired surface profile (Figure 1-1E); the desired pattern has been transferred from the mask to the film. A more complete discussion of the steps involved in microelectronics fabrication is given elsewhere [Wolf

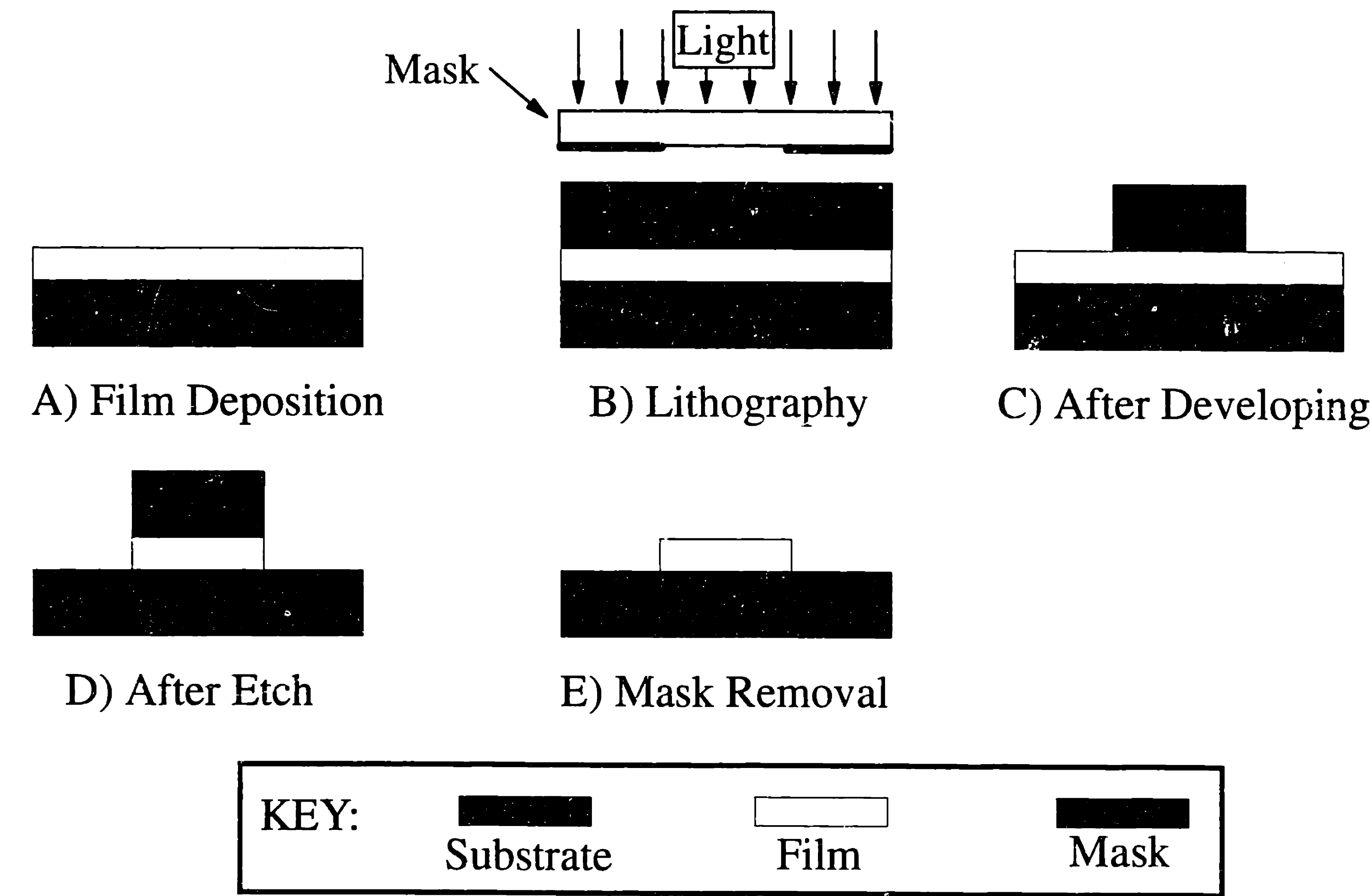


Figure 1-1: Outline of Major Steps in Subtractive Processing.

and Tauber, 1986].

1.2 Etching

Etching is critical: it determines how accurately the mask pattern is transferred to the underlying film. Etching is directional (anisotropic) when the vertical etch rate is much greater than the lateral etch rate (Figure 1-2A). When the lateral and vertical etching rates are about equal, the etching is isotropic (Figure 1-2B). Isotropically etched features are not resolved when the feature width, w, is less than twice the depth, d, of the film being etched. This is a problem because feature widths are continually decreasing with widths of 0.5 μm being common now; film thicknesses can be 0.1 to 1.0 μm, and in extreme cases for gate oxides, the film is frequently less than 0.01 μm. Isotropic etching is typical of wet etching processes; plasma or dry etching processes can be isotropic, anisotropic or a combination of the two. Below, I discuss basic plasma physics (Section 1.3), then the plasma etching of polysilicon (Section 1.4), followed by plasma-surface interactions (Section 1.5) and finally, introduce the idea of pattern dependenet etching (Section 1.6).

1.3 Plasmas

A plasma or glow discharge is a partially ionized gas that is generated within a reactor. Plasmas of interest to microelectronic processing are generated using 13.56 MHz or 2.54 GHz electric radiation; the plasmas produced using these frequencies are

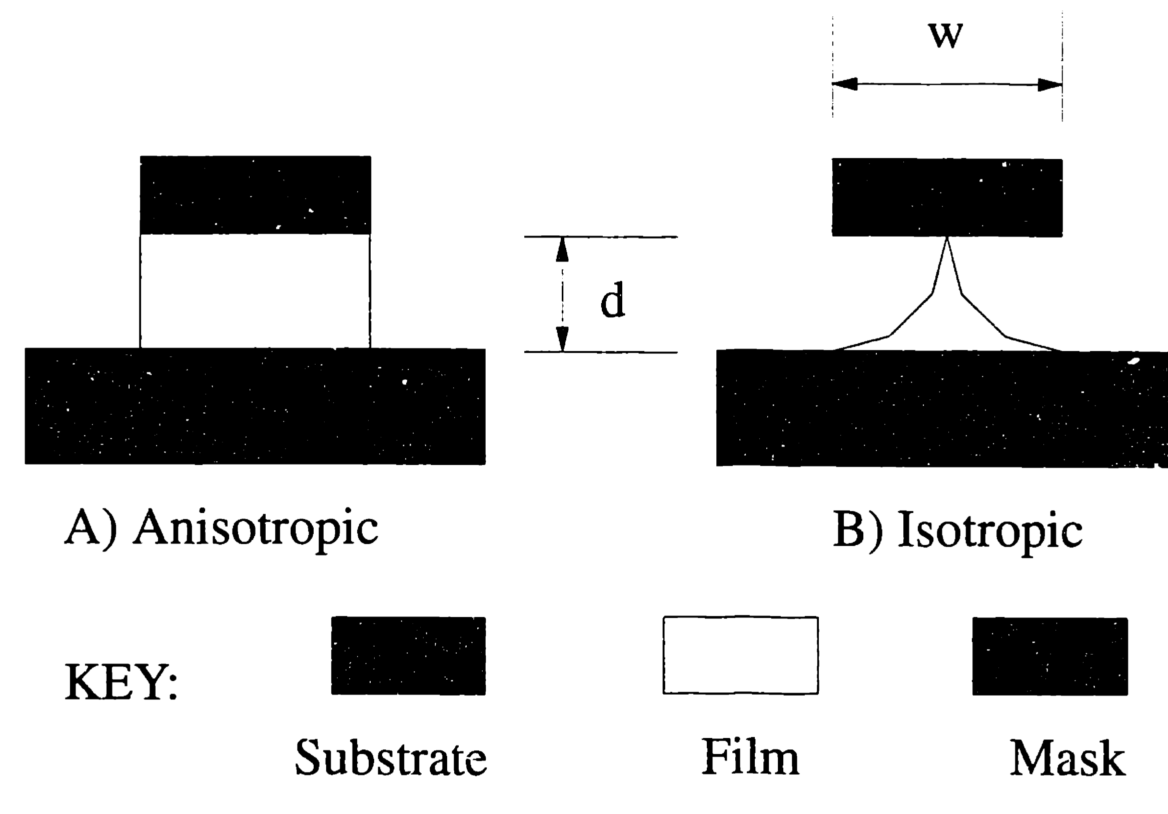


Figure 1-2: Anisotropic and Isotropic Etching Profiles

known as radio frequency (rf) and microwave plasmas, respectively. The rest of the plasma discussion will focus on rf plasmas, although it can be applied to either type. The exact method for coupling the radiation into the plasma depends upon the specific plasma source used. When the plasma is used for etching, the reactor that the plasma is generated within is known as an etcher.

The applied rf voltage is typically on the order of 100 to 1000 volts. This voltage leads to a breakdown of the gas and the generation of a plasma. Pressure inside the plasma etcher ranges from 1 mTorr to 1 Torr. This corresponds to molecular densities on the order of 10^{13} to 10^{16} molecules/cm^3, respectively. The plasma environment consists of the neutral gas or gases that are introduced into the reactor, as well as ions, electrons, free radicals and excited neutral species. All of these species are generated within the plasma by collisions. The relaxation of the excited neutrals and the subsequent emission of photons gives plasmas their characteristic glow. The fraction of the introduced gas that is ionized is typically of the order of 10^{-6} to 10^{-4} [Graves, 1989],[Chapman, 1980]. Thus, the plasma consists mainly of neutral species. Overall, the plasma is neutral, with no net charge. Electrons are much more mobile than ions in the plasma because of their significantly lower mass. Thus, they obtain energies of 1 to 10 eV while ions and neutrals typically have energies on the order of 0.025 eV (thermal energy). This imbalance between electron and ion energies means a plasma is a nonequilibrium mixture. The degree of dissociation for the feed gas (via electron impact dissociation) in the plasma is typically on the order of 10% and sometimes approaches 100% [Graves, 1989]. The high level of free radicals present makes the plasma very reactive.

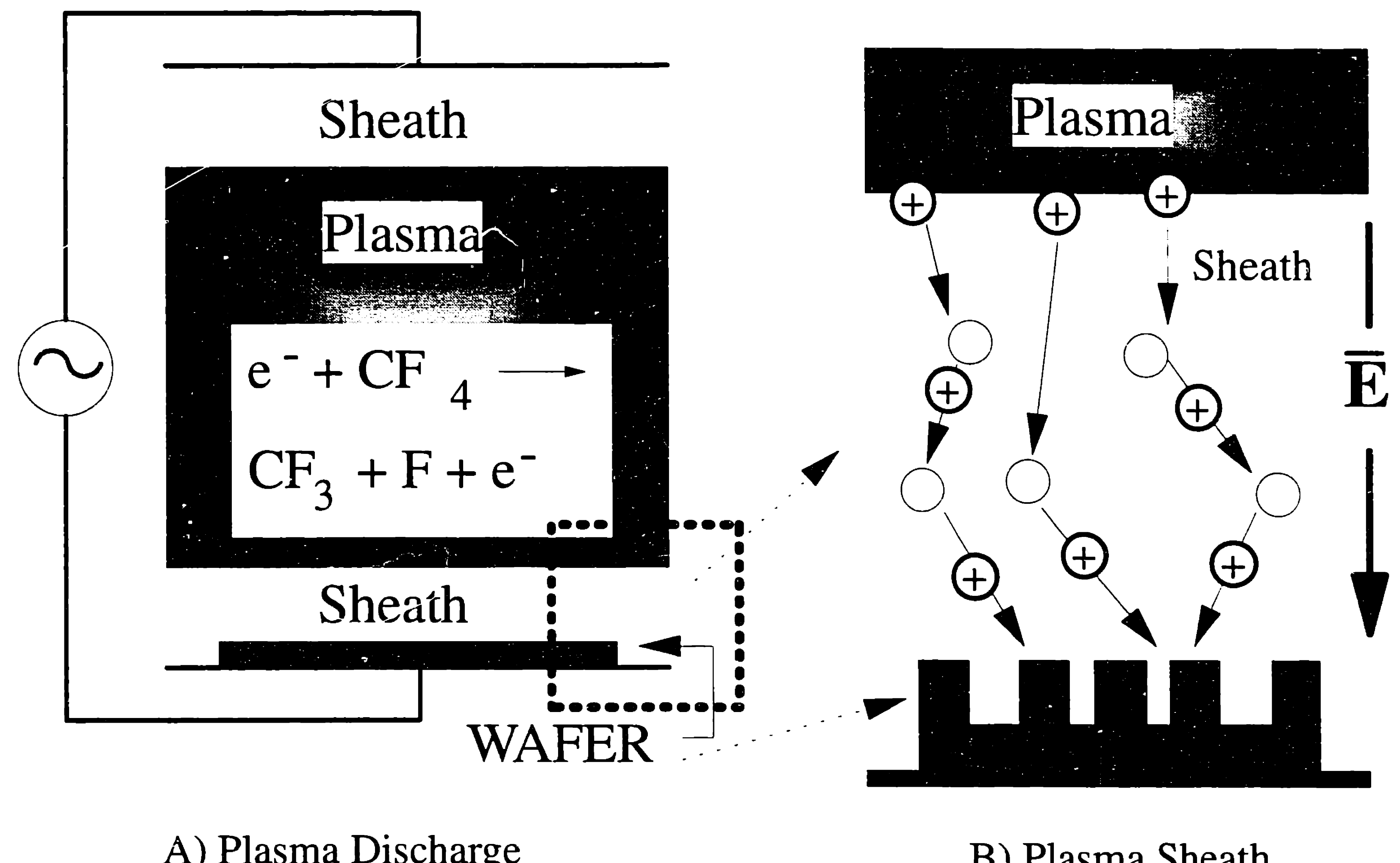
When a plasma is ignited (struck), the walls of the reactor are initially neutral.

The electron diffusivity in the plasma is 4 to 5 orders of magnitude greater than the ion diffusivity. This leads to electrons trying to rapidly escape from the plasma, while the ions lag behind. Thus, surfaces in the reactor tend to charge up negatively. The surface charge tends to repel electrons and attract ions. The charge increases until the flux of electrons and ions to the walls is net neutral.

The region immediately around the walls is known as the sheath. This region is visibly much darker than the rest of the plasma due to the lack of excitation in this region. A space charge exists in the sheath, as opposed to the plasma bulk which typically is of uniform potential. The potential difference across the sheath is on the order of 100 volts. Figure 1-3A shows a plasma and its sheaths.

The formation of the sheath is important to the interaction of plasma with a surface. Figure 1-3B shows a close-up view of the sheath region. As ions created within the plasma cross the plasma/sheath boundary, they are accelerated along the macroscopic surface normal by the electric field in the sheath, and can acquire energies on the order of 100 eV during this process. Ions collide principally with neutral molecules while traversing the sheath; ion-ion collisions are rarer because of the low ion concentration and because of Coulombic repulsion. Ion-neutral collisions give rise to variations in the ion energy and in the angles at which the ions arrive at the wafer surface. These variations are characterized by an Ion Angular Distribution (IAD) and an Ion Energy Distribution (IED), which have been modeled and measured experimentally [Liu *et al.*, 1990].

Plasma density (both ion and electron densities) can be increased with the addition of a magnetic field. The Applied Materials Precision 5000 plasma etcher (AME-5000)



A) Plasma Discharge

B) Plasma Sheath

Figure 1-3: Plasma and Its Interactions With a Surface

(Applied Materials, Santa Clara, CA) is a commercial etcher that uses a magnetic field oriented parallel to the wafer (perpendicular to the sheath electric field). In this configuration, electrons gyrate around the magnetic field lines (Larmor motion) [Mitchner and Kruger, 1973]. This increases the effective path of electrons in the plasma, decreases the electron loss rate coefficient, and increases the plasma density.

1.4 Plasma Etching of Polysilicon

Etching occurs when a plasma generates reactive species that volatilizes the surface of a film. Ion bombardment of the film also contributes to etching via physical sputtering of the film. Furthermore, Coburn and Winters [1979] have observed a synergism between reactive neutrals and ion bombardment. The etching rate with both reactive neutrals (Fluorine from XeF_2) and ion bombardment (Ar^+) was significantly greater that the sum of the etching rates with the individual sources. More recently, fundamental work was done by Gray *et al.* [1993] looking at mechanisms and kinetics for etching of Si and SiO_2 by fluorine.

Silicon used in microelectronics is either single crystal silicon or polycrystalline silicon (hereafter called polysilicon or poly). Etching of either form of silicon is expected to be similar due to the disruption of the crystal lattice at the film surface by ion bombardment. Oehrlein [1993] has used *in situ* time-resolved ellipsometry to determine that a damaged surface layer is created within seconds after plasma ignition. The thickness of this layer increases as the sheath voltage increases, indicating that ion bombardment energy plays a key role in determining the thickness. The damaged layer is

on the order of 10Å thick for a high pressure/low power (low self-bias) plasma, and is about 40Å thick for low pressure/high power (high self-bias) discharges. Consistent numbers were obtained using both ellipsometry and post-etch x-ray photoelectron spectroscopy (XPS). These thicknesses are an order of magnitude lower than the 400Å damaged layer determined in earlier work [Oehrlein *et al.*, 1984].

Polysilicon is etched with halogenated gases because of the volatile reaction products formed. Gray [1992] showed that both SiF_4 and Si_2F_6 are important reaction products for the etching of silicon with Fluorine. Unsaturated reaction products (SiF_x, x=1 to 3) become more important as the ion bombardment energy increases. Similarly, for etching with Chlorine, etch products are $SiCl_4$ at lower ion energies and $SiCl_x$ (x = 1,2 and 4) at higher ion energies [Tachi and Okudaira, 1986]. In the halogen family in general, reactivity of atomic halogen to silicon decreases with increasing atomic number, F > Cl > Br > I; the product volatility follows a similar trend [Tachi and Okudaira, 1986],[Frank and Chabert, 1993]. The data gathered by Tachi and Okudaira [1986] for the reaction of the atomic halogens (fluorine, chlorine and bromine) with silicon are shown in Table 1-1.

Table 1-1

Halogen Reaction Probabilites

	Fluorine (F)	Chlorine (Cl)	Bromine (Br)
R_{th}	$0.02 < F < 0.1$	$10^{-5} < Cl < 10^{-4}$	$Br < 10^{-5}$
$R_r = R_{200\,eV}/R_{th}$	10	$10^3 < Cl < 10^4$	$Br > 10^4$

R_{th} is the thermal reaction probability for atomic halogens; $R_r = R_{200\,eV}/R_{th}$ is the reaction proability ratio for 200 eV halogen ions to thermal atomic halogens. R_r indicates the difference between the ion-enhanced and chemical etching components. The flux of neutral particles to the wafer surface is isotropic (independent of direction), whereas the ion flux is directed. This means that a feature bottom receives a larger fraction of the ion flux than a sidewall does. Thus, R_r indicates the possible directionality of etching, with the directionality increasing with increasing R_r.

Etching of intrinsic (undoped) and n^+-doped polysilicon by atomic fluorine, chlorine and bromine proceeds at different rates. The etch rate for 5×10^{18} cm^{-3} phosphorus-doped polysilicon was 90 times greater than that for intrinsic silicon [Walker and Ogryzlo, 1991a], with identical activation energies of 28 kJ/mol. Molecular chlorine did not etch silicon at the temperatures of this study, while it did at higher temperatures and with UV assistance in other studies [Ogryzlo *et al.*, 1988]. Bromine yielded a factor of 700 difference in the etching rates between doped and undoped polysilicon, and similar studies using fluorine showed a factor of 2 or 3 enhancement [Walker and Ogryzlo, 1991a]. These results are summarized in Table 1-2.

Table 1-2

Dopant Etching Rate Effect for Halogens

Atomic Halogen ->	Fluorine	Chlorine	Bromine
$ER_{doped}/ER_{undoped}$	2-3	90	700

The reaction of atomic and molecular halogens with silicon are different under ion

bombardment (plasma conditions). Ion bombardment promotes the formation of higher molecular weight halides. Dopant effects smaller than those shown in Table 1-2 have been observed in plasmas [Flamm and Donnelly, 1981].

1.4.1 Fluorine and Chlorine

CF_4 was among the first gases used for plasma etching. The dominant reactive species is F radicals, which are lost by reaction at the wafer surface, gas phase recombination with CF_3 [Plumb and Ryan, 1986] and by surface recombination [Kiss and Sawin, 1992]. To control the stoichiometry of the plasma [Graves, 1989], various additives have been added. The O_2 is added to CF_4 to scavenge CF_x radicals, leading to an increase in the F radical concentration and an increase in etching rate [Flamm and Donnelly, 1981]. However, etching polysilicon in CF_4/O_2 discharges is isotropic [Adams and Capio, 1981] and subject to macroscopic loading effects (as will be discussed in section 1.5) [Mogab and Levinstein, 1980]. H_2 is also used as an additive, but usually for the etching of silicon dioxide. H_2 reacts with F radicals to form HF, effectively decreasing the F radical concentration, and decreases the etching rate of polysilicon while the etching rate of SiO_2 remains virtually unchanged [Chapman, 1980]. This is due to the formation of a fluorocarbon film on the surface that is volatilized by the oxygen in SiO_2 in the form of CO, CO_2 and COF_2, while the fluorocarbon film remains over the Si [Oehrlein *et al.*, 1989]. The difference in etching rates shown for Si and SiO_2 introduces the idea of etching selectivity, S, of one film with respect to another,

$$S = \frac{\textit{Etching Rate of Target Film}}{\textit{Etching Rate of Different Film}} \quad \textbf{(1-1)}$$

The selectivity of the etching process to the mask and to an underlying etch stop layer are of interest in plasma etching. Presently, for the manufacture of 16MB DRAMs, polysilicon to SiO_2 selectivities of 25:1 are needed, with that number increasing to 70:1 for 64 MB DRAMs, and projected to be 120:1 for 256 MB DRAMs [Singer, 1993].

Other fluorine sources have been used to etch polysilicon besides CF_4. The primary source used with some success has been SF_6. Thompson [1986] characterized and modeled SF_6 rf plasmas, while others have examined SF_6/O_2 mixtures [Light and Bell, 1983],[Syau et al., 1991], and $SF_6/O_2/Ar$ [Karulkar and Wirzbicki, 1989].

The combination of fluorine and chlorine chemistries has also been investigated to balance between trade-offs with each chemistry (e.g., selectivity, uniformity, anisotropic/isotropic etching profile, and loading). Among the plasma chemistries investigated are: C_2F_6/CF_3Cl [Mogab and Levinstein,1980] , C_2F_6/Cl_2 [Mogab and Levinstein,1980],[Adams and Capio, 1981], CF_3Cl [Mogab and Levinstein,1980], CF_3Cl/Ar [Allen et al., 1986a,b,c], CF_3Br [Mogab and Levinstein,1980], Cl_2/SF_6 [Guite et al. 1985],[Kimizuka et al., 1992], $SF_6/CCl_2F_2/Ar$ [Karulkar and Wirzbicki, 1989], $SF_6/CFCl_3$ [Mieth and Barker, 1983], CCl_2F_6 [Ephrath and Bennett, 1982], CCl_4/NF_3 [Bower, 1982], CCl_2F_2/H_2 and CCl_2F_2/C_2H_6 [Kimizuka and Hirata, 1985].

Chlorine chemistries have been investigated by a number of researchers in a number of plasma sources. Nguyen *et al.* [1990] investigated Cl_2/O_2 on an AME5000, Samukawa *et al.* [1990] investigated Cl_2 in an ECR, Cook *et al.* [1990] investigated Cl_2 and Cl_2/O_2 in a helical resonator source, and Sekine *et al.* [1992] investigated Cl_2 and

$SiCl_4$ in a magnetron. Others have looked at $SiCl_4$ [Tang and Wilkinson, 1991],[Curtis and Brunner, 1989], CCl_4/He [Gogolides and Sawin, 1989], Cl_2 [Richards, 1986], Cl_2/N_2 [Uetake *et al.*, 1990], $Cl_2/SiCl_4/O_2$ [Horioka, 1988], and $Cl_2/CHCl_3$ [Maa *et al.*, 1991].

1.4.2 Bromine and Iodine

Bromine-based plasma etchants have shown improvements over fluorine- and chlorine-based plasma etchants. For example, etching in CF_3Br yielded a 10% edge to center etching variation while CF_3Cl yielded up to a 50% variation. Both showed comparable selectivity and CF_3Br yielded increased directionality [Allen, 1986d]. Bromine-based chemistries show a high selectivity. Hirobe *et al.* [1987] examined deep hole etching in silicon using CF_3Br and Br_2. They observed that Br_2 discharges resulted in bowing (approx. 10°) of feature sidewalls, which they attributed to off-normal ions. In contrast, bowing was not observed in CF_3Br. They attributed to a protective film (observed by Auger electron spectroscopy (AES)) that was composed both of material sputtered off the cathode at the top of the hole, and of carbon near the bottom of the hole. Etching using a combination of fluorine-, chlorine-, and bromine-containing gases has also been reported [Yeom *et al.*, 1992].

El-Masry *et al.* [1988] investigated magnetically enhanced reactive ion etching (MERIE) using Br_2 at 2 mTorr. They reported a peak selectivity for n^+-polysilicon to oxide of 450:1 (versus 10-20 for CF_3Br). This highly selective process can be subject to micromasking by small oxide patches on the polysilicon. For this reason, an initial deglazing or brief oxide etch must be performed to remove native oxide.

Br_2 is highly corrosive; alternate sources of Br, such as HBr, are more desirable. Tsou [1989] studied HBr etching at 20 mTorr. He observed a selectivity to gate oxide of 100:1 and to photoresist (AZ-1470) of 60:1. As expected, the etching rate increased with ion energy. Minimal loading effects were observed for 0 to 60% photoresist surfaces; the etching even showed a slight inverse loading effect as the etching rate decreased with decreasing polysilicon area. However, a large loading effect was observed when the wafer was covered with 60% oxide.

Perhaps the most fundamental work to date on HBr etching has been done by Bestwick and Oehrlein [1990a]. They observed an increase in selectivity of silicon to oxide with pressure, from 13 at 10 mTorr to 400 at 100 mTorr. The silicon surface was studied by x-ray photoelectron spectroscopy (XPS). After etching, the surface contained approximately 1 monolayer of bromine, with a bromine to silicon ratio of about 2. When using an oxide mask, a thin oxide layer formed on the etched feature sidewalls; the thin oxide may play a role in yielding a directional etch due to the high selectivity of the etch. No oxide was observed on the feature bottom.

The first reports of plasma etching in iodine-based plasmas have recently been published [Fujiwara *et al.*, 1990],[Frank and Chabert, 1993]. Fujiwara et al. [1990] showed that ECR etching with HI was free of RIE Lag. Frank and Chabert [1993] showed that at pressures above 90 mTorr (and as low as 65 mTorr at low power), no etching was observed. This may be due to the very low volatility of the SiI_4 etch product. As pressure goes up and/or power goes down, the ion bombardment energy falls. The lower ion energy prevents etch products from being removed from the surface.

1.4.3 Temperature Effects

Temperature has significant effects on etching rates and selectivities. A thorough review was published by Tachi *et al.* [1991]. Tachi *et al.* [1988] investigated reduced temperature SF_6 etching of silicon at 65 mTorr. As temperature decreased from ambient to -100°C, the etching became more anisotropic and the selectivity towards oxide and photoresist increased, while the etching rate in the vertical direction was essentially unchanged; the etching rate actually increased at very low temperatures. From -100 to -130°C, very anisotropic and selective etching was obtained. Below -140°C, SF_6 condensed; this can cause the selectivity to be reduced to values far below the ambient temperature value [Bestwick *et al.* 1990b]. In Cl_2 MERIE, the SiO_2 etching rate drops drastically below 0°C, while the polysilicon etching rate does not, leading to a large increase in selectivity [Sekine *et al.*, 1988]; this has been attributed to the condensation of a protective $SiCl_xO_y$ film on the oxide. The only work to date on temperature effects in HBr etching of polysilicon has been done by Nakamura *et al.* [1988] at Fujitsu. They report anisotropic etching in the temperature range of 50 to 90 °C, with tapered etching below that. Selectivity is greatly affected by the carbon content of the plasma environment (photoresist versus oxide mask). With carbon elimination (from HBr, the piping and the reactor walls), selectivities went from 15 to 300 and up to 3000 with the addition of small amounts of oxygen [Nakamura *et al.*, 1989]. They make no indications of pattern sensitivity in HBr. Temperature effects on etching will be discussed in Chapter 6, and effects on pattern dependent etching will be discussed in Chapter 8.

1.4.4 Kinetic Studies

Kinetics for fluorine etching of silicon have been studied and reviewed by Gray [1992]. There have also been several studies of chlorine-silicon kinetics [Florio and Robertson, 1969],[Madix and Schwarz, 1971],[Barker et al., 1983],[Oostra *et al.*, 1987],[Ogryzlo *et al.*, 1988,1990], [Walker and Ogryzlo, 1991a], and [Müller-Markgraf and Rossi, 1991], and bromine-silicon kinetics [Walker and Ogryzlo, 1991b],[Tyrrell *et al.* 1989]. Diffusion coefficients have been measured for atomic chlorine and bromine in inert molecules, and in their respective molecular species[Hwang *et al.* 1987, 1989a, 1989b], [Chang *et al.* 1985].

1.4.5 Mask Material

Another phenomena that has been observed in polysilicon etching is that the mask material can affect the etch rate. During the etching of silicon with CF_4/O_2, up to a factor of 5 enhancement in the etching rate was seen using metal masks (Ag, Cu, Cr and Al) instead of a photoresist mask, and an increase in the fluorine concentration was seen using actinometry [Fedynshyn *et al.* 1989]. This phenomenon has implications on process design because of differential enhancement and depression of etch rates for different films, as has been reported for the etching of Si, SiO_2, and Si_3N_4 using Al and photoresist masks [Grynkewich *et al.*, 1990]

1.5 Plasma/Surface Interactions

The phenomena of pattern dependent etching will be discussed later, in terms of experimental observations of pattern dependencies. First, I present a more detail explanation of some phenomena in the plasma environment, in terms of plasma/surface interactions.

1.5.1 Ion Energy and Angular Distributions (IED/IAD)

The local etching rate of a thin film can be expressed in terms of a number of parameters [Gray *et al.*, 1993]

$$
\begin{aligned}
\mathrm{ER}_{\mathrm{tot}} &= f(N, J_i, E_i, E_{th}, T_s) \\
N &= \text{Neutral Flux} \\
J_i &= \text{Ion Flux} \\
E_i &= \text{Ion Energy} \\
E_{th} &= \text{Threshold Energy} \\
T_s &= \text{Surface Temperature}
\end{aligned}
\qquad \textbf{(1-2)}
$$

To determine the kinetics of fluorine etching, Gray *et al.* [1993] employed ion beams generated by either a Kauffman ion gun, or an AsTex Compact ECR ion gun. These method both yield a small energy distribution of ions. Thus, the ions appear essentially monoenergetic. In reality, the ions impinging on a surface in a plasma are distributed throughout a range of energies. Liu *et al.* [1990] measured the distribution of ion energies in an argon plasma. At 500 mTorr (the highest pressure in the study), the IED peaks at very low energy, and falls off rapidly as ion energy increases. At low pressures (10 mTorr), the peak of the distribution was shifted to a higher energy.

In addition to a distribution of energies, ions arrive at a surface with a distribution of angles. Liu *et al.* [1990] presented measurements of the ion angle distributions (IAD) for the same conditions as the IEDs. At low pressures (10 mTorr), the distribution is peaked at normal incidence. At intermediate pressures (50 mTorr) the distribution is still peaked at normal, however, the magnitude of the peak is smaller, with a significant number of ions arriving off normal. Finally, at higher pressures, the IAD is peaked off normal, at about 11 degrees.

The IED at each measured angular increment of the IAD's was measured by Liu *et al.* [1990]. At low pressure (10 mTorr), the ions arriving at normal showed an energy peak that indicated they experienced few collisions; ions arriving at angles off normal showed energy peaks at much lower energies. At intermediate pressures, the ions showed both "intermediate" behavior close to normal, and low energy peaks at higher angles. Finally, at high pressure (500 mTorr), the ions showed significant low energy peaks at all angles.

The shape of the IAD is determined by the number of collisions experienced by ions while crossing the sheath and is characterized by the scaling:

$$t_{sh} = \frac{l_{sh}}{\lambda^{+}} \tag{1-3}$$

where t_{sh} is the dimensionless collisional thickness of the sheath, l_{sh} is the sheath thickness, and λ^{+} is the mean free path for ion-neutral collisions. Low values of t_{sh} indicate that the ion-neutral collision mean free path is longer than the sheath width, so ions experience few collisions while crossing the sheath; high values indicate that the

sheath is thicker than the ion-neutral mean free path, so ions experience several collisions while crossing the sheath. These collisions lead to a loss of energy gained while crossing the sheath, and to a loss of directionality. In the work of Liu *et al.* [1990], t_{sh} was on the order of 1 at 10 mTorr and 30 at 500 mTorr. At a t_{sh} value of 10, the distributions are "fully developed" and do not change appreciable with increasing t_{sh} [Thompson *et al.*, 1988].

The IADs of Liu *et al.* [1990] have been fit to the form of:

$$\frac{di^{+}(\theta)}{dA} = c \exp\left(\frac{-\theta^{2}}{2\,\sigma^{2}}\right) \tag{1-4}$$

where $i^{+}(\theta)$ is the IAD of the ions, dA = sin(θ) dθ dϕ, θ is the angle from the macroscopic surface normal, σ is the standard deviation, c is a normalization constant, and ϕ is the azimuthal angle [Arnold *et al.*, 1993] . The value of σ for the fits was found to be 4,6, and 12 degrees at 10, 50 and 500 mTorr, respectively. IADs are shown in Figure 1-4 for $c = (\sigma\sqrt{(2\pi)})^{-1}$ (i.e., a gaussian form).

To determine the direct flux of ions, Φ_D, arriving at a point on a surface in the plasma, the IAD must be integrated over the range of "view angles" detectable by a surface feature,

$$\Phi_D = \int_{\alpha_R}^{\alpha_L} IAD(\theta)\cos(\theta)\,d\theta \tag{1-5}$$

For a plane surface, α_L and α_R are $\pm\pi/2$ and the surface samples the entire IAD. However, often features have limited view angles; in those cases only a fraction of the IAD is sampled. This is shown schematically in Figure 1-5, which shows the limiting

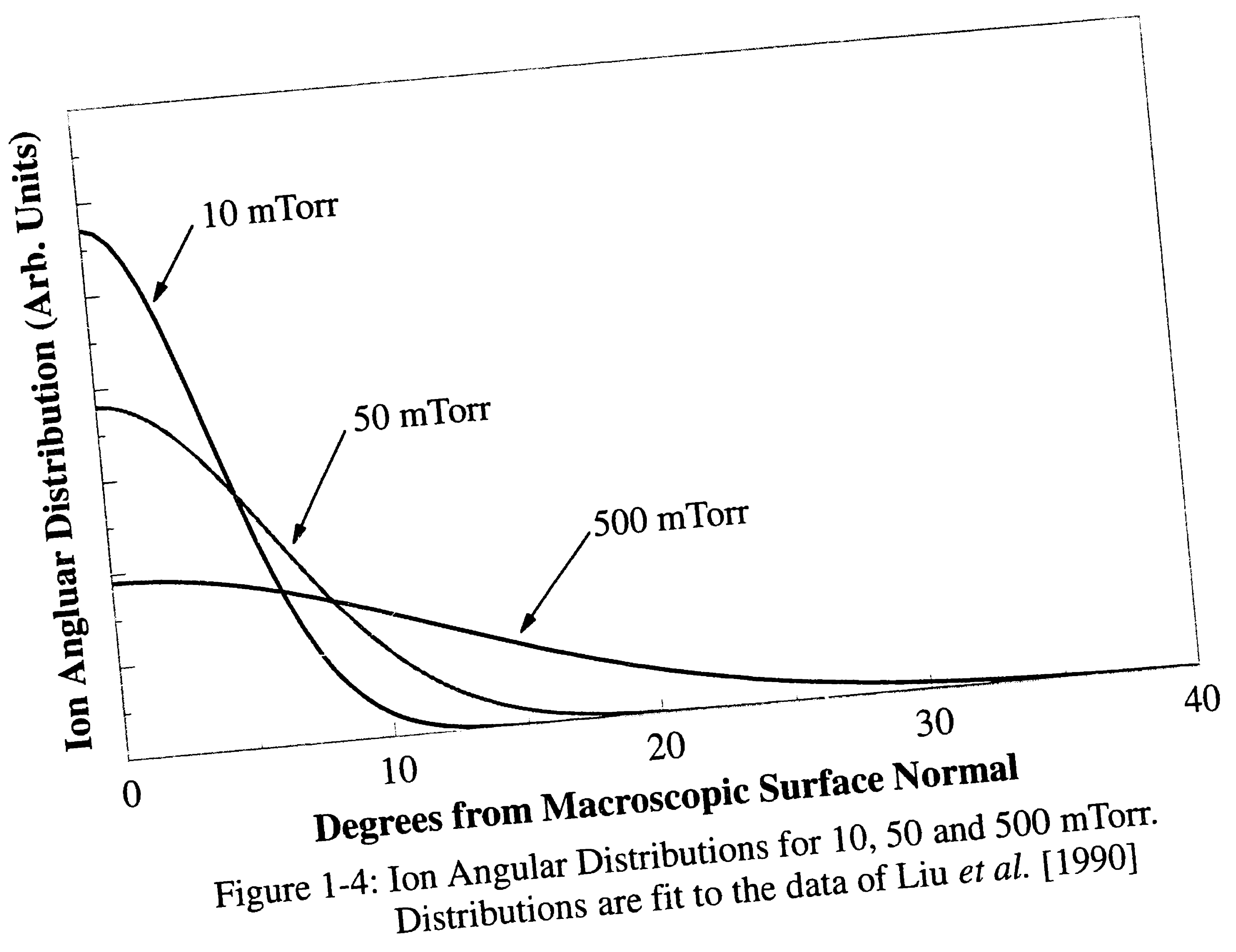


Figure 1-4: Ion Angular Distributions for 10, 50 and 500 mTorr. Distributions are fit to the data of Liu *et al.* [1990]

angular view factors for a feature on the surface, along with IADs for 10 and 500 mTorr plasmas. At 500 mTorr, α_R and α_L are within the distribution. Thus, the ion flux arriving at the point shown on the surface is below the ion flux falling on a flat surface. However, at 10 mTorr, the fraction of ions is down to near zero probability at the limiting angles. Thus, the ion flux to the feature is the same as the ion flux to a flat surface.

1.5.2 Neutral Flux

Although the above discussion shows that reaction within a feature on the surface depends on the angular and energy dependence of the arriving ions, neutral flux is also important. In general, the neutral flux to a surface is isotropic [Gottscho *et al.*, 1992]. Estimates of neutral flux to a wafer surface can be obtained from kinetic theory. The impingement flux, J_n, due to thermal motion [molecules/area-time] is given by [Chapman, 1980]

$$J_n = \frac{n\bar{c}}{4} \qquad \textbf{(1-6)}$$

where n is the gas density [molecules/volume] and $\bar{c}$ is the mean speed of molecules which can be estimated by

$$\bar{c} = \left(\frac{8k_BT}{\pi m}\right)^{1/2} \qquad \textbf{(1-7)}$$

where k_B is Boltzmann's constant, T is the absolute temperature and m is the mass of the molecule. The gas density can be estimated as

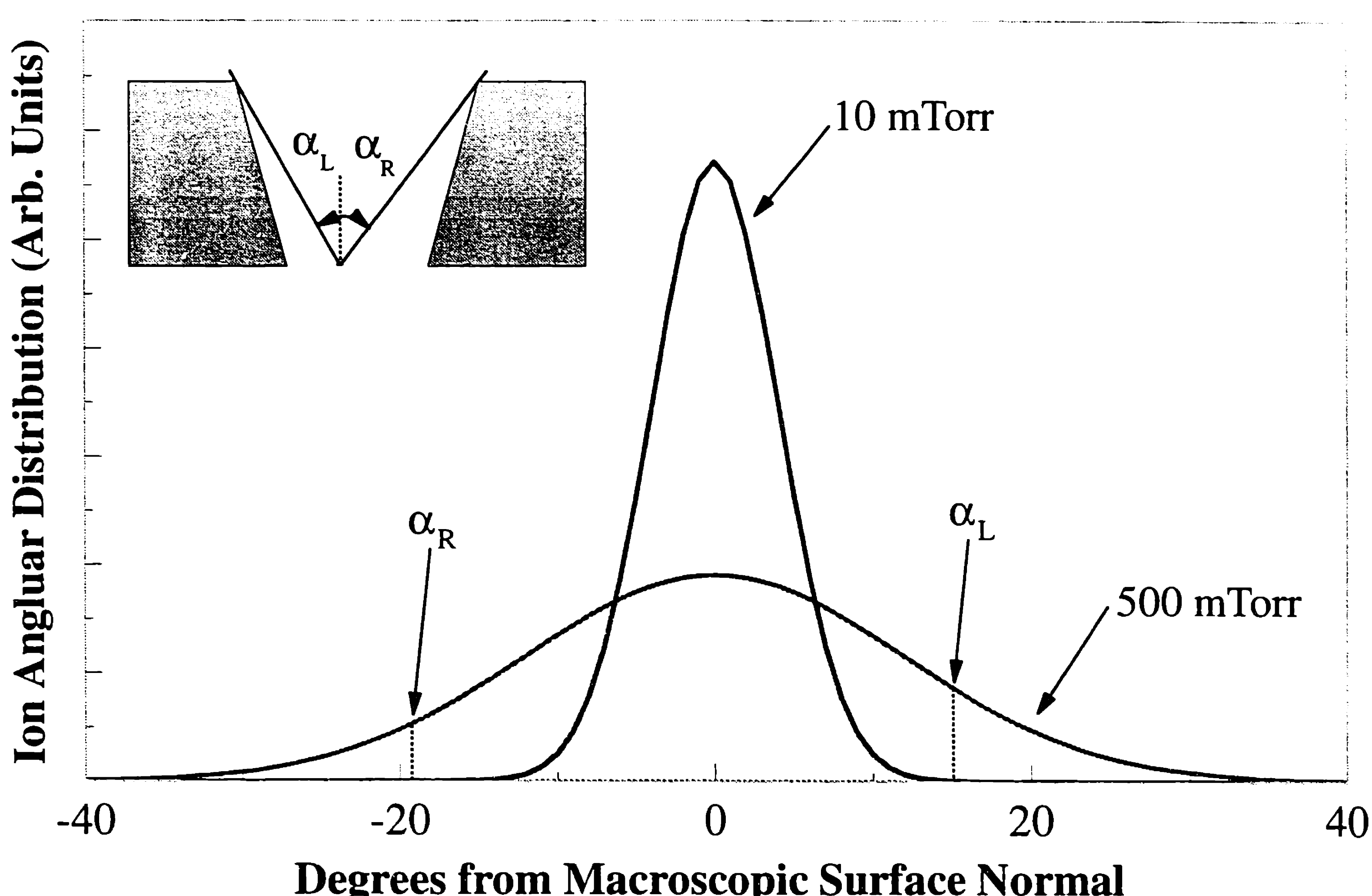


Figure 1-5: Right and Left Angular View Limits for a Feature, and Their Effect on Ion Flux for Two Different IADs.

$$n = \frac{p}{k_B T} \tag{1-8}$$

Combining these, the impingement flux of neutrals is given by

$$J_n = \frac{p}{(2 \pi m k_B T)^{1/2}} \tag{1-9}$$

1.5.3 Surface Reactions : Ion-Neutral Synergism

However, not all of the molecules that strike a surface stick. We define the sticking coefficient, γ, as the fraction of colliding molecules that stick to the surface. Assuming that a Langmuir adsorption isotherm is valid, the rate for molecular adsorption on the surface, r_a, will be given by

$$r_a = J_n \gamma (1 - \theta) \tag{1-10}$$

where θ is the fraction of the surface covered by adsorbed molecules [Smith, 1981]. This parameter θ is different from the angle θ in Equations 1-4 and 1-5. This expression for the rate of adsorption is similar to that presented by Mayer and Baker [1982]. However, they arrived at an expression for a gas that dissociates after adsorption, as shown by the stoichiometry in the following equation:

$$X_y (g) \rightarrow y X (s) \tag{1-11}$$

Their expression was the same as the right hand side of equation (1-10), multiplied by y, the number of atoms a molecule dissociates into after adsorbing. However, this is

incorrect. The correct probability for adsorption is given by $(1-\theta)^y$, not by $y(1-\theta)$ [Smith, 1981],[Adamson, 1982].

The rate for consumption of adsorbed neutrals by ion-assisted reactions, r_r, is given by,

$$r_r = k_{rxn} \theta E_i J_i \tag{1-12}$$

where k_{rxn} is a reaction rate constant, $E_i J_i$ is the ion current to the surface, and θ is as defined in Equation 1-10. Neglecting loss of neutrals from the surface by recombination reactions, a mass balance at steady state yields

$$\theta = \frac{1}{1 + \dfrac{k_{rxn} E_i J_i}{J_n \gamma}} \tag{1-13}$$

Inserting this into equation (1-12), we find the rate of reaction on the surface is given by

$$r_r = \frac{k_{rxn} E_i J_i}{1 + \dfrac{k_{rxn} E_i J_i}{J_n \gamma}} \tag{1-14}$$

Similar expressions have been derived by others [Gottscho *et al.*, 1992]. However, this is a simplification.

The total reaction rate on the surface actually includes both physical sputtering and thermal reaction of neutrals [Gray *et al.*, 1993] as well as the ion assisted reactions of equation (1-14). Given the form of the physical sputtering yield, Y_p [Steinbrüchel, 1989], the rate of physical sputtering, r_p, can be expressed as

$$r_p = Y_p \frac{J_i}{\rho} = A(E_i^{0.5} - E_{th}^{0.5}) \frac{J_i}{\rho} \tag{1-15}$$

where A is a constant, E_i and E_{th} are as in Equation 1-2, and ρ is the density of the film being removed. The thermal reaction of neutrals typically follows a form of

$$r_{th} = k_0 J_n \exp\left(\frac{-E_a}{k_B T}\right) \tag{1-16}$$

where E_a is the activation energy and k_0 is the preexponential factor. Summing equations (1-14), (1-15) and (1-16), and rearranging (1-14) we obtain an expression for the total reaction rate,

$$R = \frac{1}{\frac{1}{k_{rxn} E_i J_i} + \frac{1}{J_n \gamma}} + A(E_i^{0.5} - E_{th}^{0.5}) \frac{J_i}{\rho} + k_0 J_n \exp\left(\frac{-E_a}{k_B T}\right) \tag{1-17}$$

Gottscho *et al.* [1992] have pointed out what is termed "ion-neutral synergism" in equation (1-14), which we can also see in equation (1-17). If the ion flux goes to zero (E_i or $J_i \to 0$), then the etching rate reduces to the thermal reaction rate. Similarly, if the neutral flux goes to zero (J_n or $\gamma \to 0$), the etching rate reduces to the physical sputtering rate. When both the ion and neutral fluxes are present, a larger etching rate is obtained, as has been shown experimentally [Coburn and Winters, 1979],[Gray, 1992].

The two extremes of equation (1-17) are when the surface is saturated with neutrals ($J_n\gamma >> k_{rxn}E_iJ_i$) and when the surface is starved for neutrals ($k_{rxn}E_iJ_i >> J_n\gamma$). For

a surface saturated with neutrals, the reaction rate expression (neglecting the spontaneous etching) reduces to

$$R = k_{rxn} E_i J_i \tag{1-18}$$

where $R=R(E_i,J_i)$. While for a neutral starved surface (neglecting physical sputtering), the rate reduces to

$$R = J_n \gamma \tag{1-19}$$

where $R=R(J_n)$. We can see from (1-17), (1-18) and (1-19) that a range of possible behaviors can be observed during plasma etching, depending on the relative values of the ion and neutral fluxes to the surface. In some cases, the reaction rate will be independent of neutral flux, in other cases it will be independent of ion flux, and in other cases it will be dependent on both.

Directional etching is caused by the difference in neutral and ion fluxes to different features. A feature bottom receives a larger ion flux than a sidewall does due to the directional ion flux. The etching rate is enhanced on the feature bottom more than it is on the sidewall by the ion bombardment (Equation 1-17). Thus, the etching rate is larger in the vertical direction than in the lateral direction.

1.5.4 Neutral Transport

The expression given in Equation (1-9) for the neutral flux is valid for the random flux of neutrals striking a plane surface. When topography is added, the transport of

neutral reactive species within the surface features must be considered. The mean free path for neutrals within the gas phase is typically two orders of magnitude or more greater than the feature size. Thus, transport into the features is ballistic, with gas phase collisions being negligible. Two different transport mechanisms have been suggested. They are adsorption followed by re-emission (Knudsen diffusion), and adsorption followed by surface diffusion.

Surface diffusion has been suggested as a possible cause for microtrench formation during Si RIE in Cl_2 [Sato and Arita, 1987]. Recent modeling of surface diffusion has required unreasonably large diffusion lengths (and thus "absurd" characteristic time for reaction) to match experimental data and has led to the conclusion that transport by re-emission may be more important [Singh *et al.*, 1992]. However, such long surface diffusion lengths have been seen during deposition, but at a much higher temperature [Nomura *et al.*, 1994]. It has been shown that surface diffusion can not account for the location of microtrenches away from the sidewall [Dalton *et al.*, 1993], but that specular ion reflection can, as will be discussed in Chapter 4.

The transport of neutrals within features can occur in several different regimes (Knudsen, molecular or transition) and is characterized by the Knudsen number, which is defined as

$$N_{Kn} = \frac{\lambda_N}{d} \tag{1-20}$$

where λ_N is the neutral mean free path and d is the characteristic surface length. For RIE, $N_{Kn} \gg 1$, so that neutral transport is by Knudsen diffusion. In this regime, the diffusivity inside a pore of radius r (cm) is given by [Geankoplis, 1972]

$$D = 9.7 * 10^{-3}\ r \sqrt{\frac{T}{M}} \qquad \textbf{(1-21)}$$

where D is the diffusivity in cm^2/s, T is the absolute temperature (K) and M is the molecular weight of the diffusing species (g-mass/g-mole). The diffusivity predicted by Equation (1-21) depends only on the feature size, the temperature, and mass of the species, and not on pressure.

When a neutral species strikes a surface, it can either adsorb or scatter from the surface. Neutrals are emitted from the surface with a distribution of angles. The probability of re-emission, $P(\theta)$, is given by

$$P(\theta) = \cos^{n}(\theta) \qquad \textbf{(1-22)}$$

where θ is the angle relative to the macroscopic surface normal, and n is the emission exponent. When the re-emitted species are completely accommodated with the surface (thermal equilibrium), the exponent n becomes 1, the distribution of neutrals follows a "cosine law" [see for instance, Maissel and Glang, 1970], and the neutrals are said to be diffusely scattered. The re-emitted species can travel back into the plasma, or further into the surface feature.

The effects of Knudsen diffusion have been modeled. Arnold *et al.* [1993] used an iterative scheme to model the transport effects in the Si/F system. They observed that the effect of feature aspect ratio on neutral transport is more important at higher reaction probabilities due to depletion of reactants at the trench bottom. Singh *et al.* [1992] have modeled the effect of Knudsen diffusion on profile evolution, where they were able to obtain good agreement between model results and experiments using reasonable

parameters.

An extreme case of neutral transport has been termed "Neutral Shadowing" by Gottscho *et al.* [1992]. In this limiting case, the etching rate is considered to be neutral limited (following Equation (1-19)); neutrals arrive at the surface with an isotropic distribution, and stick to sidewalls with unity probability (no scattering). In this case, only neutrals with a direct line-of-sight to the feature bottom will arrive there to contribute to etching. Their analysis yielded

$$\frac{R_b}{R_f} = \sin\left[\arctan\left(\frac{1}{2A}\right)\right] \tag{1-23}$$

where R_b/R_f is the ratio of the etching rate at the bottom center of the trench to the etching rate for a flat region and A is the feature aspect ratio (etch depth / line spacing). This equation predicts a very rapid decrease in etching rate with aspect ratio due to neglecting the flux to the sidewalls. Other simulations predict a much slower effect of aspect ratio on etching rate [Arnold *et al.*, 1993].

Coburn and Winters [1989] have analyzed neutral transport in terms of vacuum conductance in high aspect ratio features, both in terms of product transport away from the feature bottom and reactant transport to the feature bottom. They determined that the impedance to product removal in a hole with A=10 etching at 1 μm/min was still three orders of magnitude too small to be important. Their analysis of reactant transport showed that conductance could be important at higher reaction probabilities. However, they assumed the ion flux was monodirectional, neglecting the effect of the IAD on etching rate in high aspect ratio features.

Thermal neutrals should have an isotropic distribution [Gottscho *et al.*, 1992].

However, "fast" neutrals created by charge exchange collisions within the sheath can have a significant angular dependance. Typical charge exchange collisions are

$$O^+_{2(fast)} + O_{2(thermal)} \rightarrow O_{2(fast)} + O^+_{2(thermal)}$$
$$Ar^+_{(fast)} + Ar_{(thermal)} \rightarrow Ar_{(fast)} + Ar^+_{(thermal)} \quad \textbf{(1-24)}$$

Numerical studies of fast neutrals in Ar indicate that they can contribute significantly to the sputter yield for $t_{sh} > 1$ (Equation 1-3) [Manenschijn and Goedheer, 1991]. A significant number of neutrals with high energy were found to arrive with angles of between 0 and 20 degrees.

1.6 Pattern Dependent Etching

Simply put, "pattern dependent" (or pattern sensitive) etching is the dependence of the etching profile, rate, selectivity, or uniformity on the surface topography being etched. Pattern sensitivity has been a recognized problem for a number of years [Mogab and Levinstein, 1980]. These effects are becoming more important as industry desires to use one etching tool to produce multiple products, ASICs (application specific integrated circuits), without reoptimizing the etching process to each new product. The effects of pattern dependencies are more drastic at submicron linewidths. They can occur on two different scales, global or local.

Global pattern dependencies are observed as differences in etching across an entire wafer, or on a die, as uniformity differences or loading effects. In this case, the etching behavior is characterized by a Damköhler number of the second kind, Da_{II}, which

is the ratio of the characteristic reaction velocity (ϑ_r) to the characteristic diffusion velocity (ϑ_D) and can be expressed as

$$Da_{II} = \frac{\vartheta_r}{\vartheta_D} = \frac{r}{D / l} = \frac{rl}{D} \qquad \textbf{(1-25)}$$

where l is a characteristic length for diffusion (one-half the wafer diameter for full wafer effects, one-half the die size for die effects), D is the diffusivity of reactants (in the gas phase), and r is the surface reaction rate coefficient. Large values of Da_{II} indicate reactant transport is much slower than reaction, while small values indicate that reactant transport is much faster than reaction.

Local pattern dependencies are microscopic effects, occurring as line of sight processes at the wafer surface due to the mean free path for gas phase collisions being several orders of magnitude larger than the etch feature dimensions. These local dependencies are usually observed with the aid of scanning electron microscope (SEM) photos. Examples of several pattern dependencies are shown in Figure 1-6. One observation is that smaller features etch slower than larger features (Figure 1-6A), and has been called "RIE Lag." A second type of pattern dependency is characterized by a curvature of the sidewalls (Figure 1-6B), and is called "Barreling" or "Bowing." A third type of pattern dependency is the formation of trenches, or etch pits, deeper than the surrounding etch plane (Figure 1-6C). This is referred to as "Trenching" or "Microtrenching."

Several possible physical mechanisms may cause these pattern dependent etching

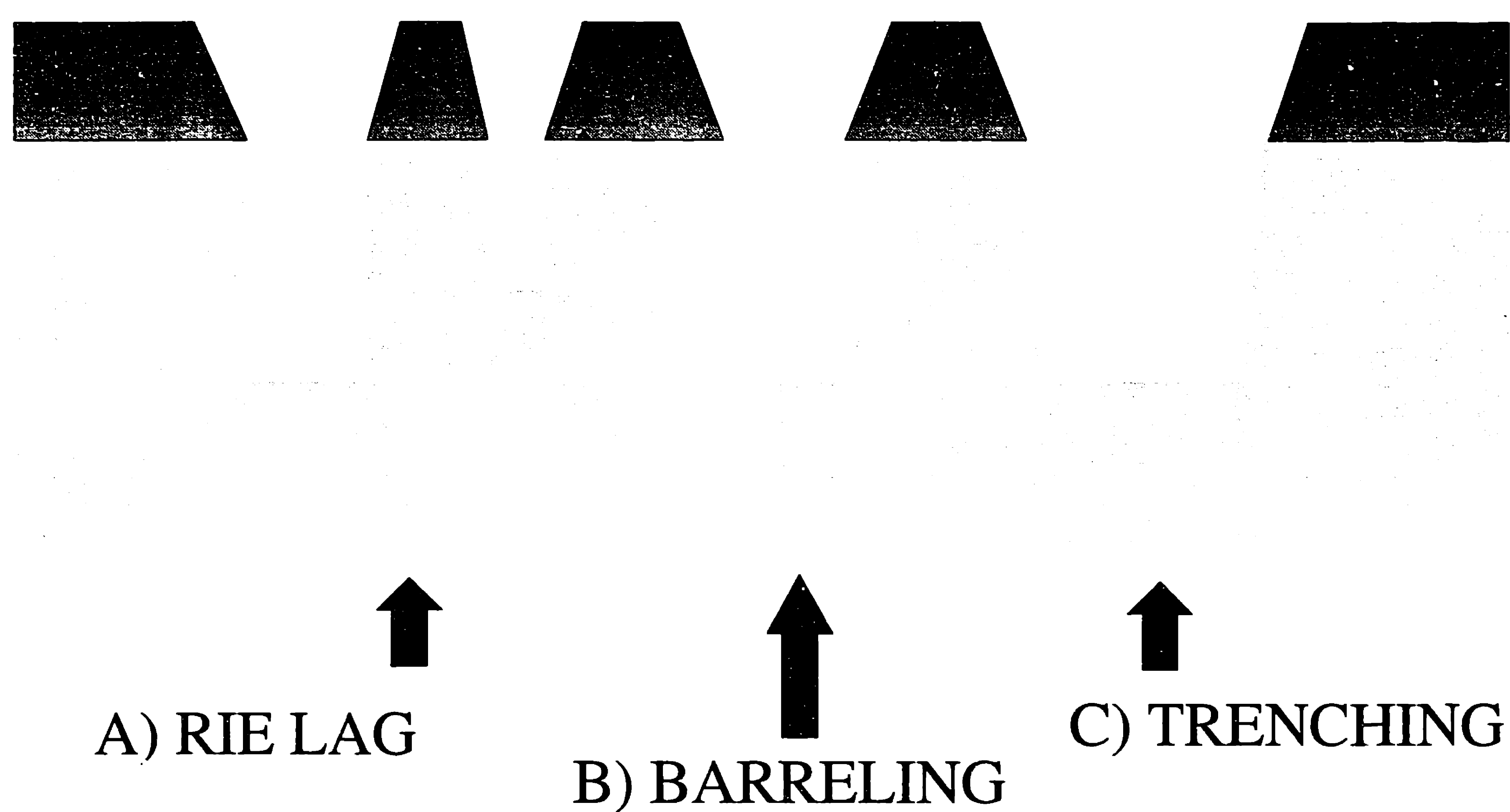
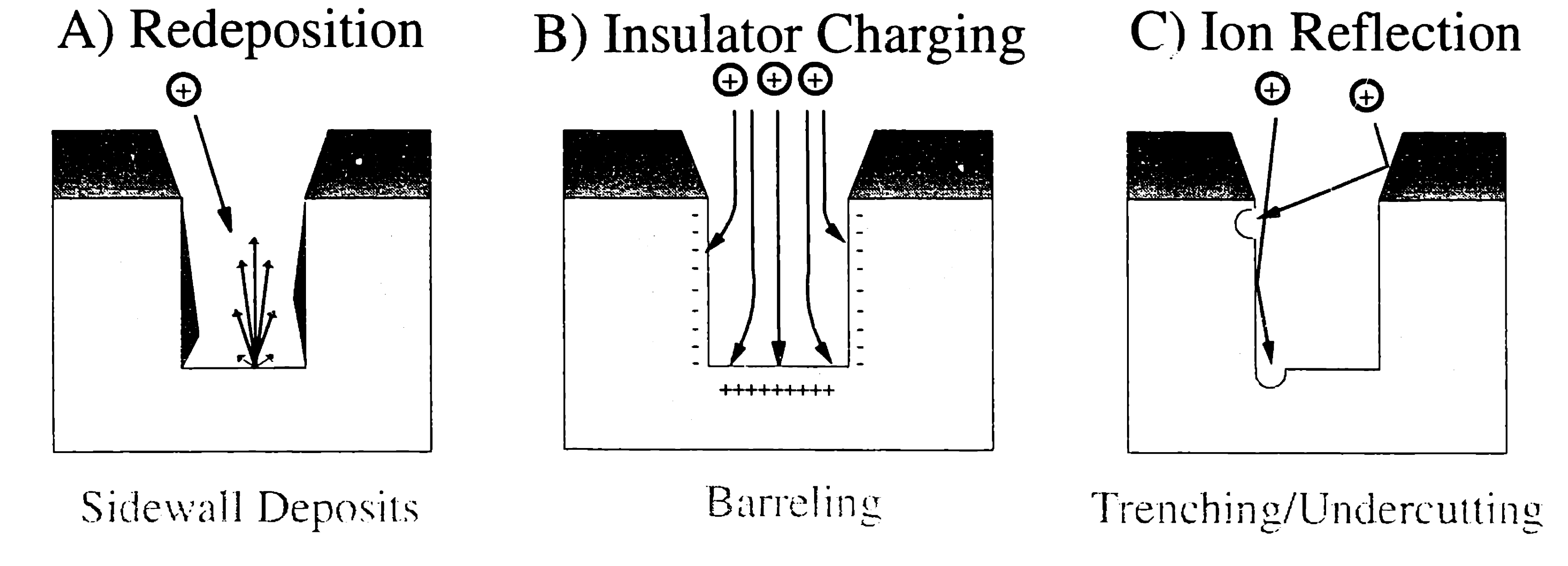


Figure 1-6: Some Typical "Pattern Dependent" Etching Phenomena

phenomena. Some of them are shown in Figure 1-7. Redeposition of sputter etched products can be responsible for deposits onto sidewalls, altering the etching characteristics (Figure 1-7A). Charging of insulating layers can lead to deflection of ion trajectories within a feature, leading to bowing (Figure 1-7B). Reflection of ions off of features can locally enhance the ion flux and the etching rate leading to the formation of trenches (Figure 1-7C). Differences in feature size lead to different "view angles" that determine what fraction of the IAD a feature will receive, leading to RIE Lag (Figure 1-7D). A number of examples of these phenomena are given below.

1.6.1 Undercutting Variations With Line Spacing

As mentioned in section 1.5.4, transport of neutrals into features by Knudsen diffusion can be important. Peng *et al.* (1986) studied the etching of submicron polysilicon lines (0.1, 0.2, 0.27, 0.38 and 0.5 μm mask openings) with trilayer masks (0.6 μm PMMA resist + 700 Å Si + 1 - 2 μm AZ1470) in CCl_4 plasmas. The mask was not undercut for a 0.5 μm gap. However, smaller openings showed undercut, with the greatest undercut occurring at the smallest gap spacing. They attributed these observations to two possible causes. First, CCl_2 (formed by electron impact dissociation of CCl_4) may stick to or react with the trilayer resist sidewalls, and in a narrow gap, the concentration of CCl_2 may be sufficiently depleted so that a passivating polymeric film does not deposit, resulting in undercut. A second explanation is that the trilayer resist may charge, influencing the trajectory of "incident radicals." However, this seems unlikely as perturbations in the electric field would influence the trajectory of ions and



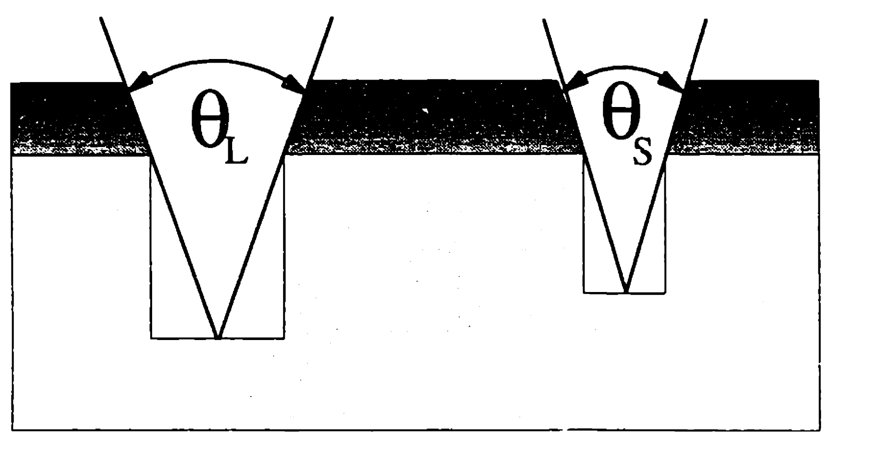


Figure 1-7: Possible Physical Causes of Some "Pattern Dependent" Etching Phenomena

not radicals.

1.6.2 Aspect Ratio Dependent Etching

Aspect ratio dependent etching (ARDE), RIE Lag or "Pattern-Factor Dependent Etching" is the generalization that trenches (or holes) with a smaller spacing (diameter) etch slower than features with larger spacing (diameter) [Singh *et al.*, 1989]. This phenomena has been observed in a wide range of plasma chemistries for many different types of films. These include SiO_2 with CF_4/O_2 and CF_4/He [Singh *et al.*, 1989], HPR 204 (a novolac-based photoresist) with O_2 [Janes and Pilz, 1993], Si with CF_3Br/SF_6 [Lee and Zhou, 1991], Si with CCl_2F_2 [Chin *et al.*, 1985], n^+ poly-Si with Cl_2 and HCl [Fujiwara *et al.*, 1990], poly-Si with Cl_2 [Fujiwara et al., 1989], and GaAs with $Cl_2/SiCl_4$ [Cooper *et al.*, 1987]. A thorough review article has recently been published on ARDE [Gottscho *et al.*, 1992]. Chin *et al.* [1985] observed that the etching rate depended solely on the aspect ratio of the features, independent of the opening size, and decreased almost linearly with increasing aspect ratio.

There are a number of possible causes for ARDE. As discussed in sections 1.5.1 and 1.5.4, the feature size (aspect ratio) may limit the arrival of angularly dependent fluxes. ARDE is often attributed to such 'shadowing' effects. ARDE will be discussed in more detail in Chapter 8.

1.6.3 Microtrenching, Notching

Microtrenches and other notches have been observed on both feature bottoms and sidewalls during etching and ion-beam milling. In ion-beam milling, a beam of energetic ions is used to physically etch a surface. It has been observed that trenches are formed on feature bottoms at the base of photoresist sidewalls during ion-beam milling [Glöersen, 1975],[Lee, 1979]. Simple ion reflection models can explain the observed trenches. [Younger and Haynes, 1982].

Microtrenching has been observed on feature bottoms during the etching of polysilicon over oxide with Cl_2 [Nguyen *et al.*, 1991], in Si etching with Cl_2/BCl_3 [Ameen et al., 1988], and in Si etching with Cl_2, Cl_2/H_2, and $Cl_2/SiCl_4$ [Sato and Arita, 1987]. Notches have also been observed in sidewalls [Ameen *et al.*, 1988], [Ohki et al., 1987]. All of these features could be explained by ion reflection off of sidewall features. Although not specifically addressed, microtrenches are visible in other systems such as etching of SiGe/Si in HBr [Bestwick et al., 1991]. A model has been developed for ion reflection and is presented in Chapter 4.

1.6.4 Other Microscopic Pattern Dependencies

Bowed or barreled sidewalls have been observed during plasma etching [Hirobe *et al.*, 1987]. Several researchers have proposed that charging of insulator surfaces (mask or film) may result in a potential that is large enough to deflect the ion trajectory. [Economou and Alkire, 1988],[Arnold and Sawin, 1991]. The simulations of Arnold and Sawin [1991] represent "worst" case scenarios and indicate that in some cases, charging can be important.

Perhaps one of the most important factors in determining etch directionality is plasma polymerization (deposition of a passivating film on the surface being etched) and/or redeposition (deposition of etch products on the surface being etched). Fluorocarbon films control the etching rate of silicon in a CF_4/H_2 plasma, where the etching rate was found to be inversely proportional to the film thickness for films over 10 Å [Kay, 1984]. The passivating surface films are responsible for directionality: films deposited on sidewalls are subject to less energetic ion bombardment, and thus remain on the sidewalls, acting as a diffusional barrier both for reactive species to diffuse in and for volatile etch products to diffuse out; films deposited on the bottom of a feature will be removed by the significantly greater ion flux impinging on the bottom of a feature. [Adams and Capio, 1981]. Adams and Capio [1981] observed the effects of carbonaceous film deposition while etching polysilicon in C_2F_6/CF_3Cl plasmas. A 1:1 mixture resulted isotropic etching, while a 4:1 mixture yielded anisotropic etching. The only difference in the two cases was the carbon to fluorine ratio and thus, the tendency to form polymer. The effect of redeposition on profile directionality through surface passivation has been successfully modeled [Jackson and Dalton, 1989]. Other attempts have only addressed deposition rates, without considering passivation [Lehman *et al.*, 1977],[Lii and Jorné, 1990]. Excessive deposition is unwanted as it leads to cleaning problems after the etch. Redeposition from a photoresist mask results in anisotropic etching with non-carbon containing etchants such as BCl_3, Cl_2, and HCl [Allred *et al.*, 1989]; etching without an organic mask (i.e., SiO_2) produced "extreme" undercut, but etching with a photoresist mask produced an anisotropic etch. Plasma polymerization is an isotropic process in which deposition is limited only by transport of film precursors

while redeposition is a line of sight process in which deposition is a function of geometry.

1.6.5 Full Wafer Uniformity

One global pattern dependent etching phenomenon of interest is radially varying etching rates. In multiwafer batch etchers, it was recognized that wafers experienced a radially nonuniform etching rate. The most frequent observation was that the etching rate at the edge was greater than at the center due to consumption of reactants across the wafer being greater than the reactant transport rate ($Da_{II} > 1$). This led to what was referred to as a "bullseye" clearing pattern [Nagy, 1984]. This same phenomenon is also observed in single wafer etchers of current interest. Current methods to measure uniformity across a wafer consist of measuring initial film thickness at a number of points, partially etching the wafer and remeasuring film thickness after etching. This is slow and of limited value. In Chapter 5, I describe the invention of a new *in situ* monitor to measure plasma etching uniformity during etching. This has great potential use as a process development tool and as a process control diagnostic.

1.7 Goals of This Research

This research focuses on the pattern dependencies in the plasma etching of polysilicon. The goals were to (1) increase the understanding of pattern dependent etching phenomena, (2) develop *in situ* control monitors for etching to help characterize pattern dependencies, and (3) to develop processes that are pattern independent.

The work presented here first details the installation of a commercial plasma etching tool (AME-5000), and the addition of plasma diagnostics to this tool in Chapter 2. Chapter 3 deals with the characterization of plasma etching processes via a neural network analysis. Chapter 4 details the modeling of microtrench formation during plasma etching of a polysilcion gate over a thing gate oxide. Chapter 5 details the invention and development of an *in situ* monitor to determine full wafer uniformity. Chapter 6 details reduced temperature plasma etching. Chapter 7 details the measurement of spatially varying wafer surface potentials during plasma etching. Finally, Chapter 8 details the investigation of ARDE using the technique detailed in Chapter 5.

Chapter 2

Experimental Apparatus and Diagnostics

An essential part of investigating etching processes is that the results be directly applicable to others. In the past, much research was done using "homemade" plasma etchers (see, for example, [Richards, 1986]). An attempt has recently been made to produce a standard research reactor, both for experiments and for comparison to modeling efforts; this is known as the Gaseous Electronics Conference (GEC) Reference Cell. An other approach is to use an industrial etcher so that results are directly transferable. However, this is usually prohibitively expensive; industrial etchers currently cost between $1 x 10^6 and $2 x 10^6.

In 1990, a state-of-the-art industrial etcher (an Applied Materials Precision 5000 (AME-5000) plasma etcher) was donated to MIT by the manufacturer. All of the plasma etching experiments done as part of this research were done on this etcher.

The AME-5000 is a single wafer etcher; it consists of a central load lock chamber that connects to up to 4 etching chambers. Our etcher (System #6166) used 100-mm wafers and housed two different etching chambers. One for polysilicon etching (Chamber B) and one for silicon dioxide etching (Chamber A). An elevator that holds up to 8 wafers is located within the load lock chamber. Outside of the load lock are two wafer cassettes that each hold up to 25 wafers. An overhead view of the AME-5000 indicating all of these features is shown in Figure 2-1.

The polysilicon etching process investigated on this etcher used a mixed

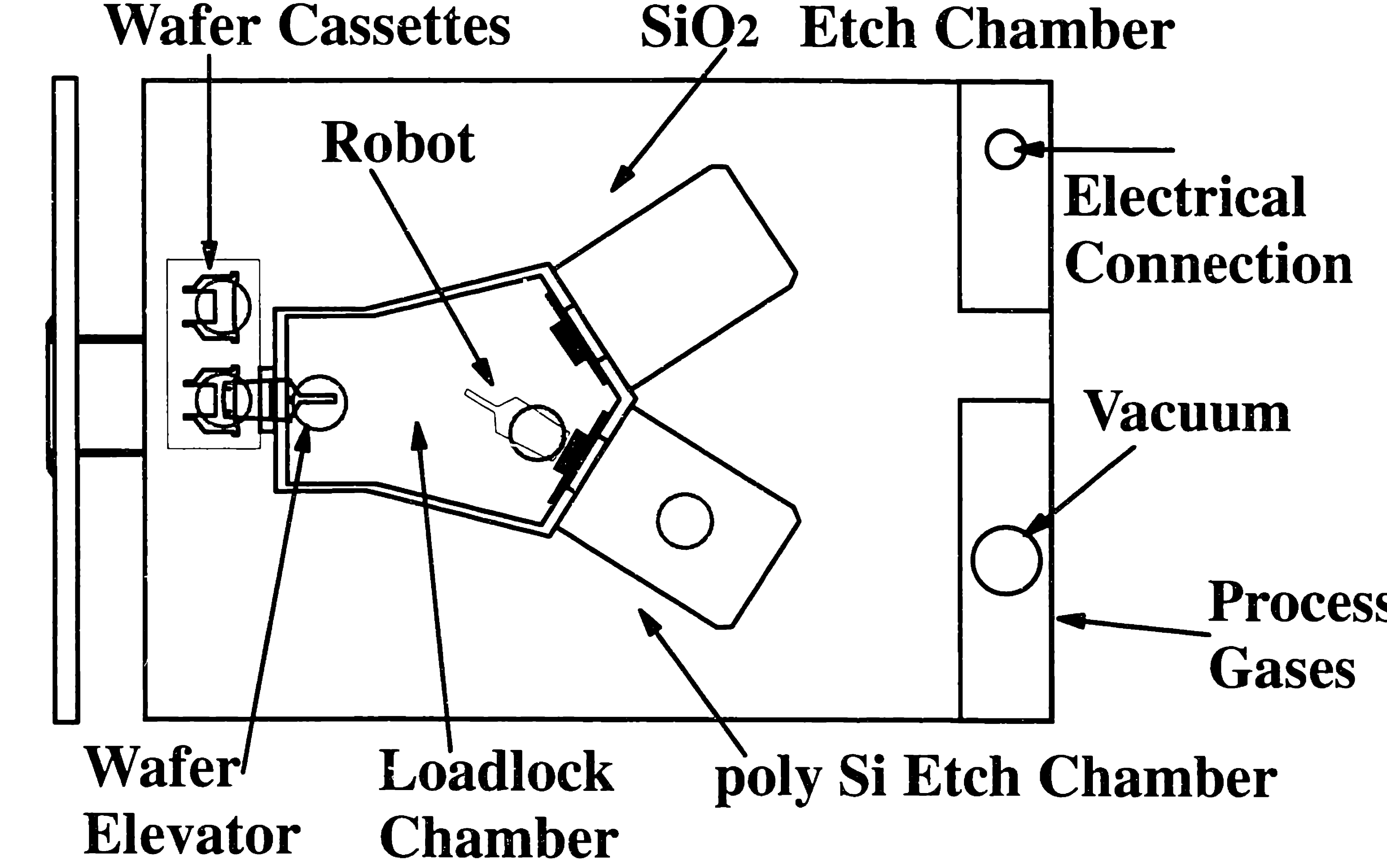


Figure 2-1: Overhead View of AME-5000 With Major Components Indicated

chlorine/bromine chemistry. The "standard" etching process was: 30 SCCM Cl_2, 10 SCCM HBr, 100 mTorr, 250 W, 75 Gauss.

2.1 Installation of AME-5000

An AME-5000 is typically installed into a manufacturing environment where the proper facilities (gas handling, chilled water, vacuum exhaust, pneumatic air, house N_2) exist to support the machine. Lab 66-225, where the AME-5000 was installed, was not equipped to support the machine. For this reason, a number of facilities were added to the lab to support the AME-5000. Details on the facilities are given in the following sections.

2.1.1 Gas Handling - Etchants, Pneumatic Air, and Purge N_2

A number of etchant gases were available for use on the AME-5000. Each chamber had a total of 5 mass flow controllers (MFCs) (Unit Instruments Inc., Orange, CA). In addition to this, two extra MFCs on an external manifold could be used on either of the chambers. A list of the etchant gases available, the chambers they could be used on and the full scale flow ranges are listed in Table 2-1.

Table 2-1

Etchant Gases and Flowrate Ranges on AME-5000

Gas	Chamber	Flow (SCCM)	MFC
He (90%)/ O2 (10%)	B	18	1
HBr	B	100	2
Cl_2	B	87	3
NF_3	B	144	4
SF_6	B	300	5
CHF_3	A	10	8
CF_4	A	126	9
O_2	A	100	10
Ar	A	140	11
$C_2F_4H_2$	A	92	12
Ar	A/B	20	E
Ar	A/B	1000	E

A standard etchant gas delivery system was established. The basic configuration is shown in Figure 2-2. An assembly was constructed that mounted onto the cylinder head (shown inside the dotted lines on Figure 2-2). The assembly consisted of a regulator (Matheson (Gloucester, MA) 3803 series for corrosive gases, 3102 series for non-corrosive gases), diaphragm valve (Nupro SS-DSV51) to a double end shut off quick disconnect fitting leading to a pump/purge assembly, a 0.02 μm particle filter (Matheson 6190-V4FM) and two more diaphragm valves; one valve to lead to a gas distribution line for room 66-221 and one lead to a panel-mounted final valve that fed into the gas distribution box on the AME-5000. All connections were made with VCR (metal gasket) fittings.

The purge assembly consisted of two separate diaphragm-valved lines: a N_2 line

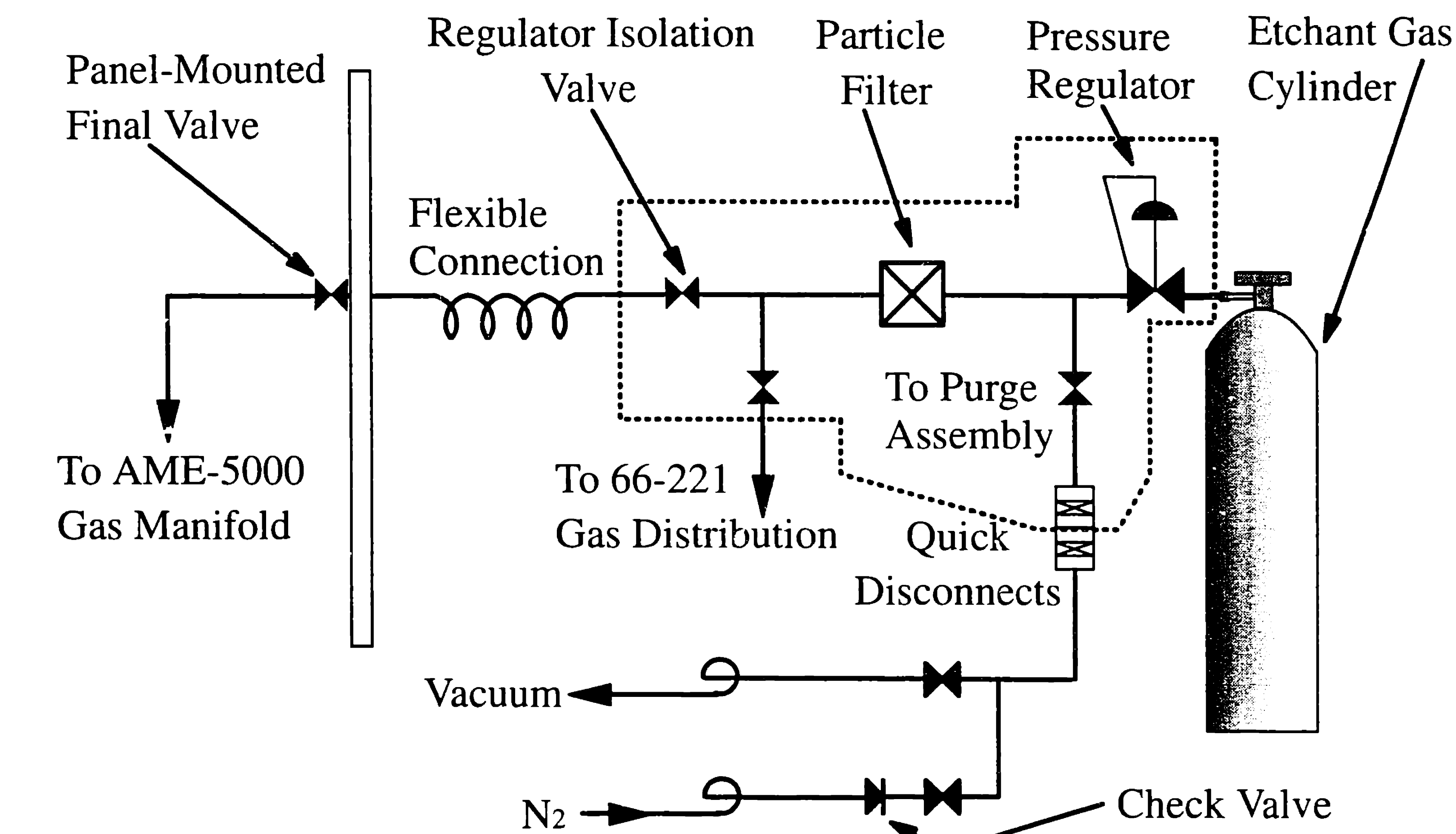


Figure 2-2: Standard Etchant Gas Handling System

and a vacuum line. The N_2 line had a check valve (Nupro SS-4C-VCR-1) to protect against contamination of the N_2 source. A liquid nitrogen (LN_2)-cooled sorption pump (for example, Huntington (Mountain View, CA) #SP-151) was used to supply the vacuum. The sorption pump was regenerated by heating up to room temperature.

For toxic gases, all of the components visible in Figure 2-2 were contained within an exhausted hood. The gas distribution lines leaving the hood and running to the AME-5000 were all welded stainless steel (0.250 inch O.D., 0.035 inch wall, 16 rms or better electropolished, Stainless Pipe (Cambridge, MA)).

The purge assembly was used to purge the regulator assembly and gas delivery lines before gas service was begun. This was accomplished by a two part procedure. The first part was "cycle purging" of the entire assembly. The regulator delivery pressure valve was closed completely (highest delivery pressure) so that the space up to the cylinder head would be purged. With the delivery valve at the AME-5000 closed, N_2 was flowed into the regulator assembly and gas lines, and then valved off. The N_2 was removed using the sorption pump; this N_2 purge and pump constituted one "cycle". The process was repeated up to 30 times for the initial service of a gas line, and up to 10 times after a line was exposed to air. After the initial cycle purge, a second purge was done. During this, N_2 was flowed with the delivery valve on the AME-5000 open so that gas flowed into the etcher. Starting at the regulator and working down line to the AME-5000, a heat gun was used to warm up the lines to aid in the desorption of water. Following these cleanings, a line considered to be ready for gas delivery service and was not exposed to air again, except for cylinder changes.

The AME-5000 has a large number of pneumatic valves requiring 60 psi air to

function. The standard compressed air service in Building 66 was 40 psi and fluctuated significantly. We installed a 100 psi compressed air line into the lab and regulated it down to 60 psi for use on the AME-5000.

Large quantities of N_2 were required for turbo pump purges, roughing pump purges and ballast, and for chamber venting. No house N_2 was available. We installed a N_2 gas distribution system using a Western Enterprises (Westlake, Ohio) "Accu-Trol" LC-7-2 manifold. Two LN_2 dewars (230 liters each) provided up to 5030 ft^3 each of N_2 gas.

2.1.2 Other Utilities

Cooling water was needed for several units on the AME-5000: Neslab chiller, heat exchanger, radio frequency (RF) generators, and mainframe (magnets, and turbo pumps). The maximum flow requirements are listed in table 2-2.

Table 2-2

AME-5000 Cooling Water Requirements

Unit	Max Flow (GPM)
Neslab	2
RF Gen-A	1
RF Gen-B	1
Heat Exchanger	10
Main Frame	10

Cooling water was provided from a central closed-loop chilled water system. Water was fed to a manifold panel, where it was regulated (Spence Engineering Co. (Walden, NY) #D36) to 60 psi and then filtered through a 10 μm filter element (Cole Parmer (Niles, IL) #G-1509-07), with a bypass loop for continuous service during a filter change. The water was split among 5 flowmeters for distribution to the various units of the AME-5000 that required cooling water. Each line consisted of a shut off valve and a flow meter. The manifold was constructed using sweated copper pipe. Water return to the chilled-water system was done at three different locations (close to the point of use). Each return line had a check valve on it to prevent reverse water flow and a shut off valve.

A PVC vacuum trunk was build to provide exhaust for the vacuum pumps. Each chamber was pumped with a 150 liter/sec turbomolecular pump (Alcatel 5150) backed by a Leybold (Export, PA) D25BCS roughing pump, using perfluorinated pump oil for corrosive gas service. The load lock was pumped by a Leybold Trivac D30A. Exhaust flow rates for the pumps, the gas panel on the AME-5000, and for the hood housing the corrosive gases were established by the MIT safety office.

2.2 Plasma Diagnostics

Plasma etching processes are inherently complicated. A number of process monitors or "plasma diagnostics" have traditionally been used to monitor the plasma etching process, both to gain insight into the physical processes occurring in the plasma

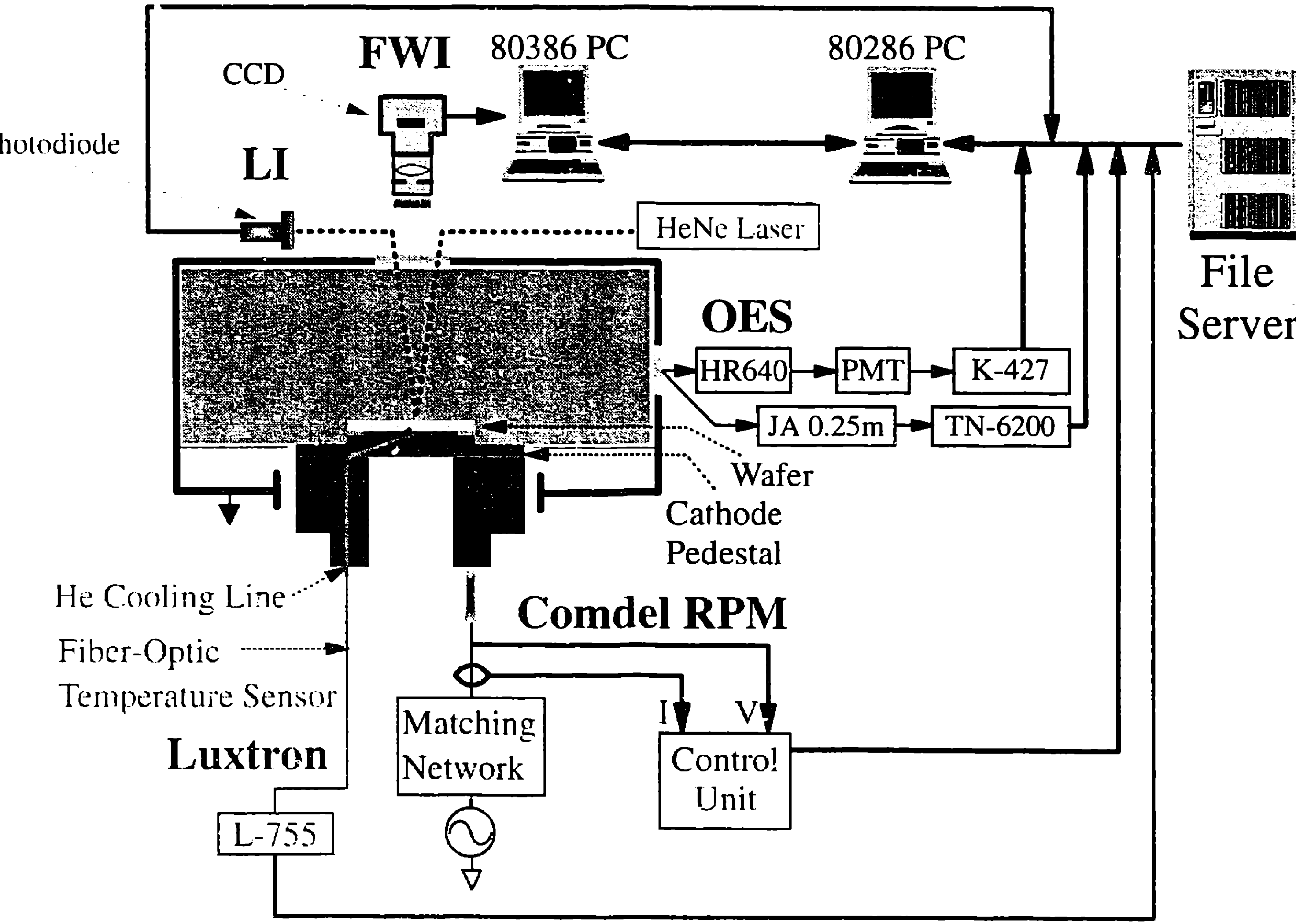


Figure 2-3: Plasma Diagnostics Installed on the AME-5000

and to observe end-point or clearing of a film within the plasma. Diagnostics can be divided into "invasive" and "non-invasive" categories. Invasive diagnostics are those that are inserted into the etching chamber and may possibly perturb the plasma while non-invasive diagnostics are those which do not have to be inserted into the plasma, such as optical techniques. A number of plasma diagnostics were added to the AME-5000 and are shown in Figure 2-3. These diagnostics will be discussed in greater detail in the following sections.

2.2.1 Laser Interferometry

One of the most obvious changes occurring during etching is the removal of a film from a wafer surface. This removal process is characterized by the rate of removal, the etching rate. A simple non-invasive method to measure the etching rate *in situ* is laser interferometry (LI).

In LI, a laser beam is bounced off of a wafer surface. The beam both reflects off the wafer surface and is transmitted through the film and reflects off the lower interface. As the wafer etches and the film thickness changes, the reflected beams go through periodic maxima and minima due to the interference of the two beams. The etching rate can be calculated for near normal incidence as

$$ER = \frac{\lambda}{2n\,\delta t} \tag{2-1}$$

where ER is the etching rate, λ is the wavelength of light used, n is the index of refraction for the film being etched, and δt is the time difference between adjacent maxima or

adjacent minima. The theory of interferometry is discussed in much more detail in Chapter 5.

A sample LI signal and its second derivative are shown in Figure 2-4. For the first 55 seconds, 5000 Å of polysilicon is being etched (5500 Å/Min). From approximately 55 seconds to 140 seconds, 1000 Å of silicon dioxide is etched (700 Å/Min). Finally, After 140 seconds, the silicon wafer is being etched. From the above estimates of etching rate, we obtain a poly:oxide selectivity of 8:1. The second derivative is plotted here because it is often useful for end-point determination in addition to the actual signal [Collot *et al.*, 1991].

Our LI system consisted of a Uniphase HeNe 0.95 mW helium-neon (HeNe) laser operating at 6328 Å. The laser beam is passed through a neutral density filter (d = 2.0) to reduce the signal strength to avoid saturating the detector, bounced off a mirror (Melles Griot, #02MLE003), through an optical port in the top of the AME-5000 and onto the wafer. The reflected beam is brought out of the etcher, through a 6328 Å optical bandpass filter (Edmund Scientific, (Barrington, NJ) #G30,701), and into a Devar 529-2-5 integrated optical detector (Devar Inc, (Bridgeport, CT)). The photodiode output voltage was measured using an A/D converter as will be discussed in Section 2.3.1. A schematic for the amplifier used to measure the LI signal is shown in Figure 2-5.

At 6328 Å, the indices of refraction for polysilicon and silicon dioxide are 3.882 and 1.457, respectively (see Table 5-1). These correspond to a film thickness ($\lambda/2n$) of 815 Å for poly and 2172 Å for oxide removed between adjacent minima on the interferometry plot. The "wiggles" visible on Figure 2-4 (*e.g.* during the oxide portion of the etch) are due to the rotating magnetic field on the AME-5000 causing fluctuations in

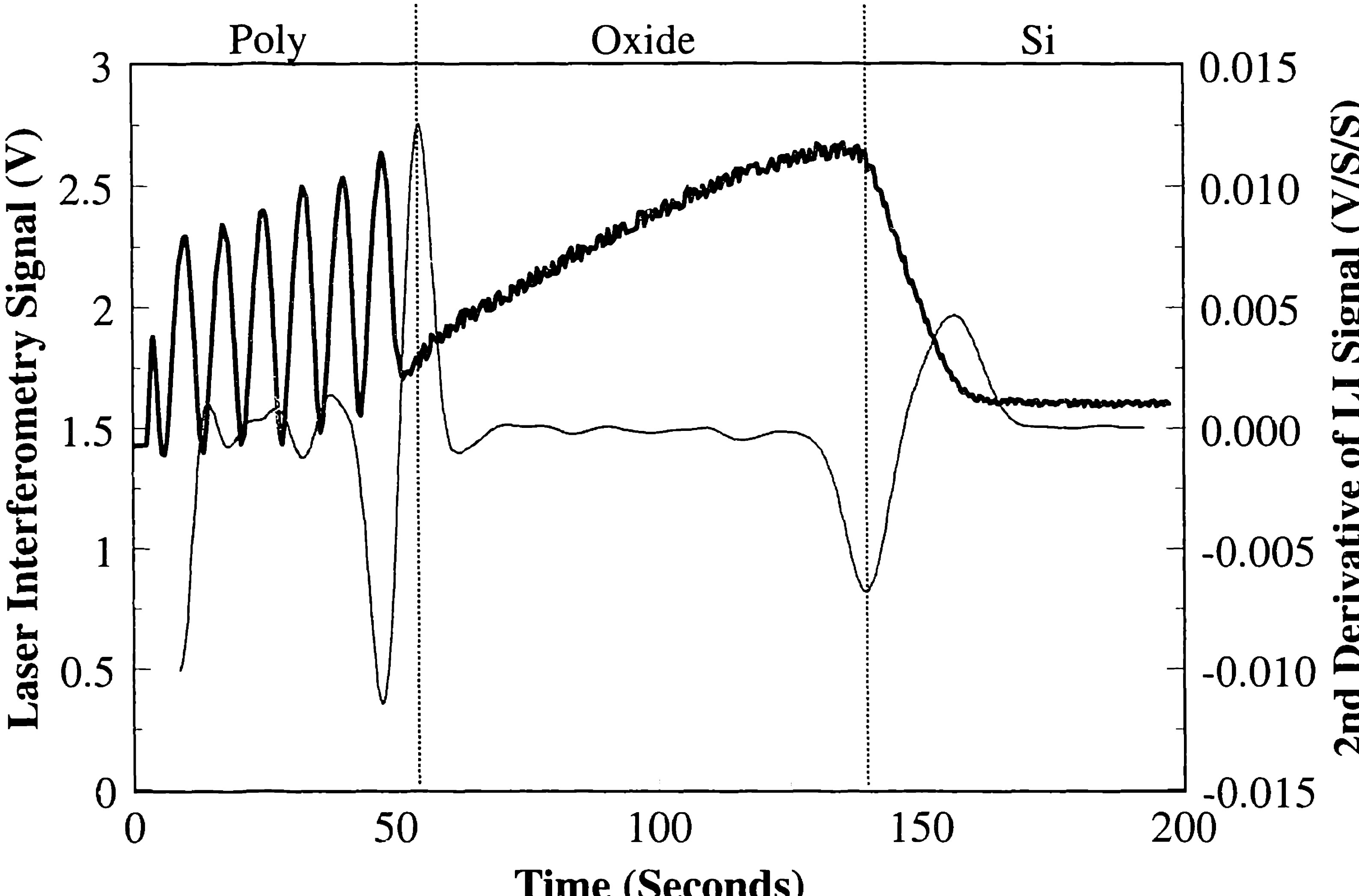


Figure 2-4: Sample Laser Interferometry Signal, and its Second Derivative. Etch is Through Poly, Oxide and Into Si Wafer.

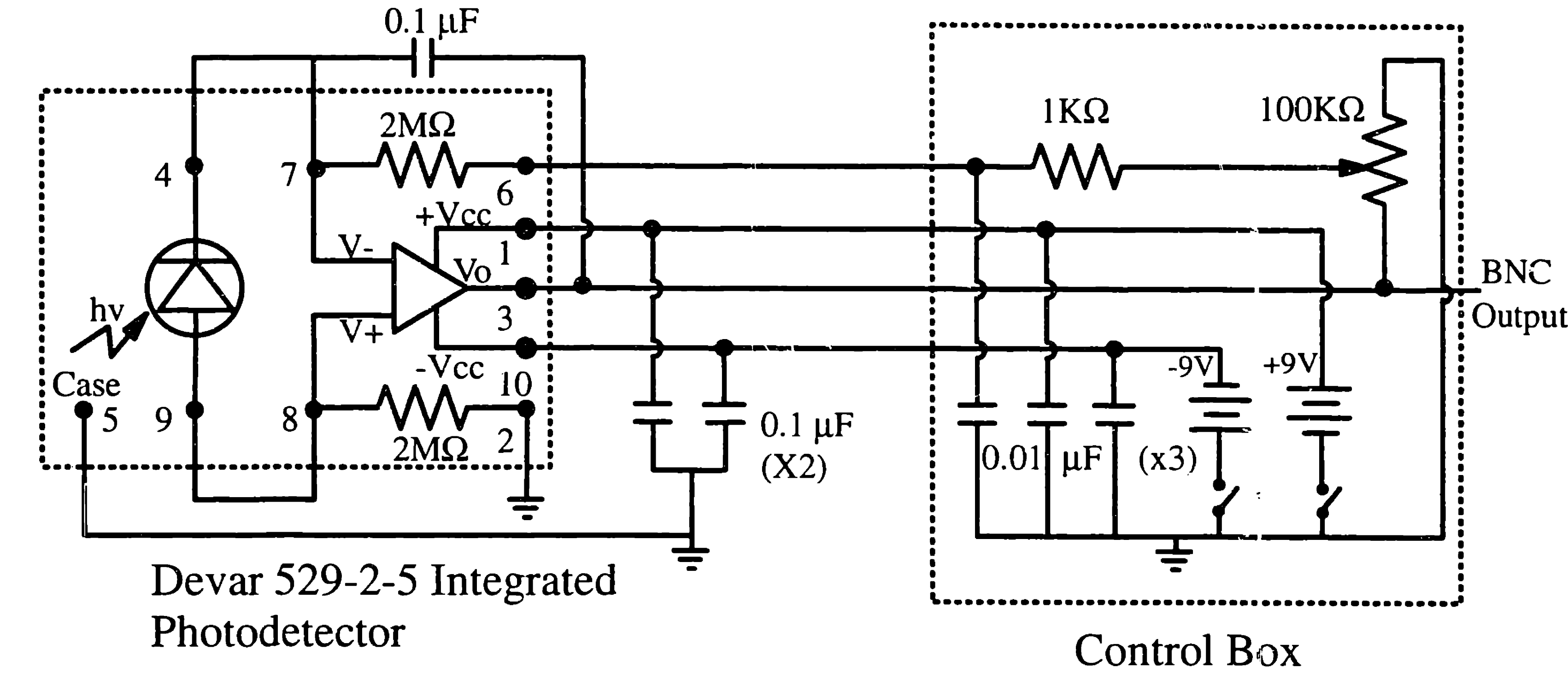


Figure 2-5: Laser Interferometry Photodetector Amplifier Circuitry.

the plasma optical emission. Some of the optical emission falls within the bandpass of the optical filter so that the observed signal is a combination of the laser signal and the plasma optical emission signal. The plasma component of the signal is very small and does not interfere with the measurement; the intensity of the plasma component can be reduced by moving the optical detector further away from the etcher.

A key limitation of laser interferometry is that the etching rate determined is for a single point on the wafer. The etching rate at multiple points (etching rate uniformity) is a very useful value. A new and novel solution to this problem is discussed in Chapter 5.

2.2.2 Optical Emission Spectroscopy

One of the most obvious things about a plasma is that it glows. Species within the plasma are excited by electron impact collisions; they subsequently relax, emitting a photon. The wavelength of these photons are characteristic of the species from which the light originates. The emission intensity depends on a number of parameters [Rossnagel and Saenger, 1989],

$$I^x_{emis} = n_e n_x A \int \rho_e(E)\, v\, \sigma_x(E)\, dE = A\, n_x n_e f(T_e, \sigma_x) \qquad \textbf{(2-2)}$$

where I^x_{emis} is the emission intensity for the species of interest, x, n_e is the electron density, n_x is the density of species x, A is a geometry dependent constant, E is the electron energy, $\rho_e(E)$ is the electron energy distribution, v is the electron velocity, $\sigma_x(E)$ is the cross section for excitation, $k_B T_e$ is the electron temperature, and $f(T_e, \sigma_x)$ is the excitation rate for species x.

A non-invasive plasma etching diagnostic would be to monitor the optical emission of a plasma. A monochromator is an instrument used to measure optical emission intensity as a function of the emission wavelength. Light enters the monochromator, is directed onto a diffraction grating where it is dispersed into its component wavelengths, and then exits them monochromator to enter a detector. The theory of monochromators is described in much more detail elsewhere [Pierson, 1979]. Three different monochromators with two different detectors are described next.

A) Xinix

Two small monochromators came with the AME-5000 (Xinix, Inc. (Santa Clara, CA) #01-1004-15). They were 100-mm focal length f/3.5 scanning monochromators with a resolution of 20 Å, and a range of 2000 - 8000 Å using a 1200 groove/mm diffraction grating, with a Hamamatsu R928 photomultiplier tube (PMT) detector. A scan of a HBr/Cl_2 plasma (10 SCCM/30 SCCM, 100 mTorr, 250 W, 0 Gauss) taken with this monochromator is shown in Figure 2-6; several of the emission lines here were identified and are indicated on this plot. These monochromators were self calibrating, leading to repeatable assignment of wavelengths.

B) Jarrell Ash

A 275-mm focal length f/3.8 Jarrell-Ash (Fisher Scientific Co., (Springfield, NJ)) Monospec-27 monochromator was also installed; it had a resolution on the order of

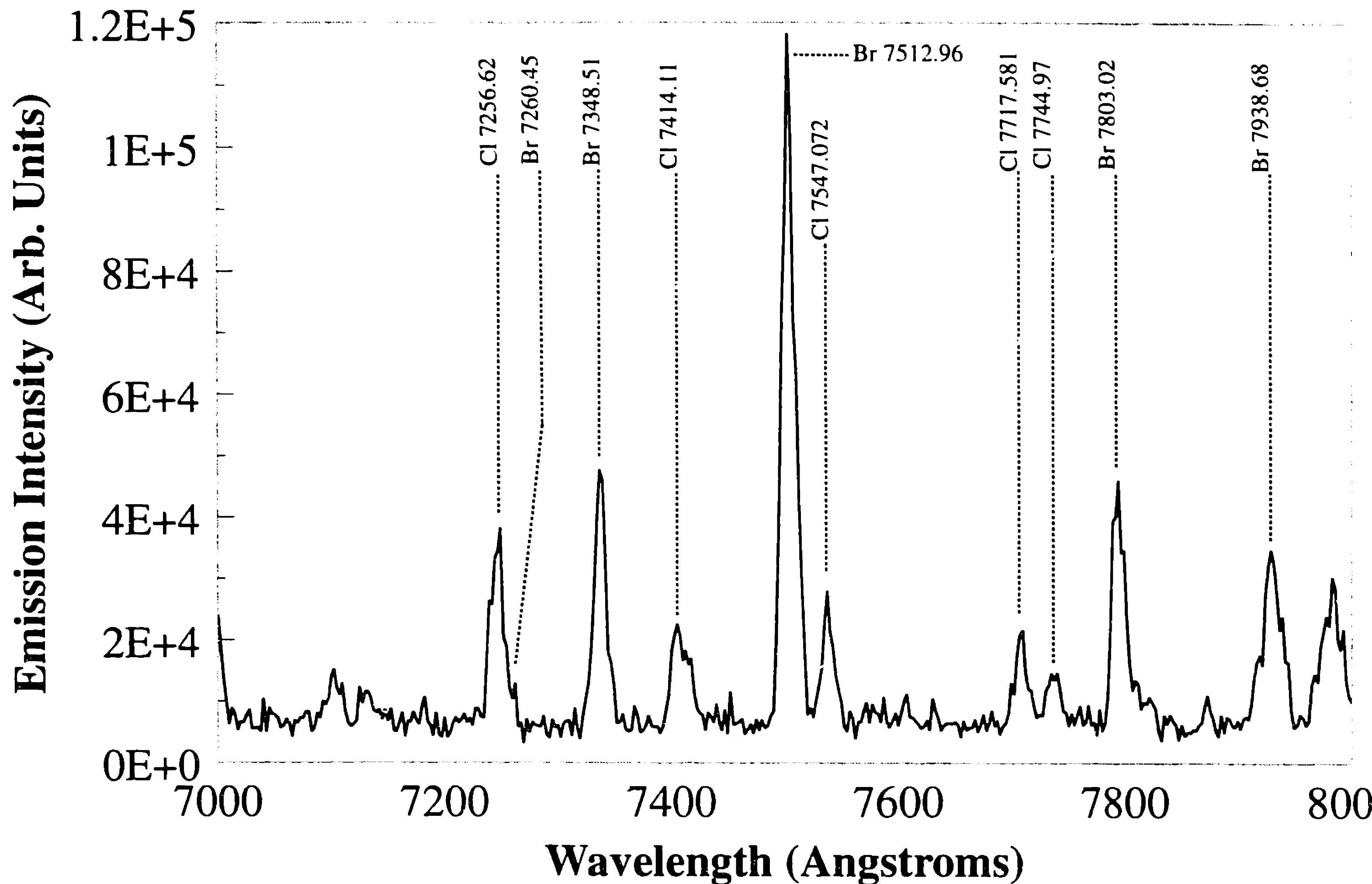


Figure 2-6: Plasma Optical Emission Spectrum for HBr/Cl_2 Plasma Taken with Xinix 100-mm Scanning Monochromator.

5-10 Å. Three different diffraction gratings were available; they are listed in Table 2-3.

Table 2-3

Diffraction Gratings on Jarrell-Ash Monospec-27

Grating Number	Grating Type	Density (groove/mm)	Blaze (Å)	Range (Å)	Dispersion (Å/mm)
1	Ruled	1200	3000	1900-6000	3.0
2	Holographic	150	4000	2000-8000	24.0
3	Ruled	1200	5200	3800-9000	3.0

Grating 3 was the most frequently used because of the high resolution (high groove density) and because of the large useable range.

A TracorNorthern (Middleton, WI) TN-6112 DARSS (Diode Array Rapid Scan Spectrometer) with a TN-6200 computer interface was used as the detector. The TN-6112 is a 1024 element linear diode array (LDA). The advantage of the DARSS is that the monochromator functions as a spectrograph; the LDA allows for the simultaneous measurement of intensity over a large wavelength space while a PMT allows the measurement of a single wavelength. The first and last 4 elements of the LDA were not used because they were unreliable. Approximately 600 Å of wavelength space could be viewed using a 1200 groove/mm grating (#1,3), and approximately 480 Å of wavelength space could be viewed using a 150 grove/mm grating (#2).

The gain variation across the LDA was ±22% (typically ±11%) [TracorNorthern, 1982]. The effect of this variation was clearly visible when looking at the dark signal

(background signal). The dark signal is plotted in Figure 2-7 as the black line. This reading was taken with the plasma off, integrating for the same amount of time as would be done for an actual reading with the plasma on. The background reading changes as a function of position on the LDA. Also shown on this figure is a reading taken with the plasma on (10 SCCM HBr, 30 SCCM Cl_2, 100 mTorr, 250 W, 0 G). Peaks are visible on this spectrum that are not visible on the background spectrum. Several peaks visible on both spectra (*e.g.*, 7040 Å and 7095Å) are not real peaks, but are artifacts of the gain variation in the LDA. The actual signal spectrum is obtained by subtracting the background spectrum from the raw signal spectrum (Figure 2-8). This spectrum can be compared to that of Figure 2-6. An improvement in peak resolution is easily visible. The DARSS spectrum only covers 7000 Å to 7600 Å because that is the range visible in a single setting.

Signals were taken with the DARSS by "integrating" the acquired light for a period of time. The spectra of Figures 2-7 and 2-8 were acquired using a 2000 ms integration time. When using a rotating magnetic field in the AME-5000, the signal integration times were chosen to be whole integer multiples of the magnetic field rotation time.

The Jarrell-Ash monochromator was not self calibrating. With the "play" in the gearing on the handle to move the grating, an absolute wavelength offset of ±50 Å was not unusual. This was corrected for by adding a constant offset to the wavelengths to move the emission lines to fall at the proper location. The value of this offset was computed by examining the location of several known lines for an Ar plasma.

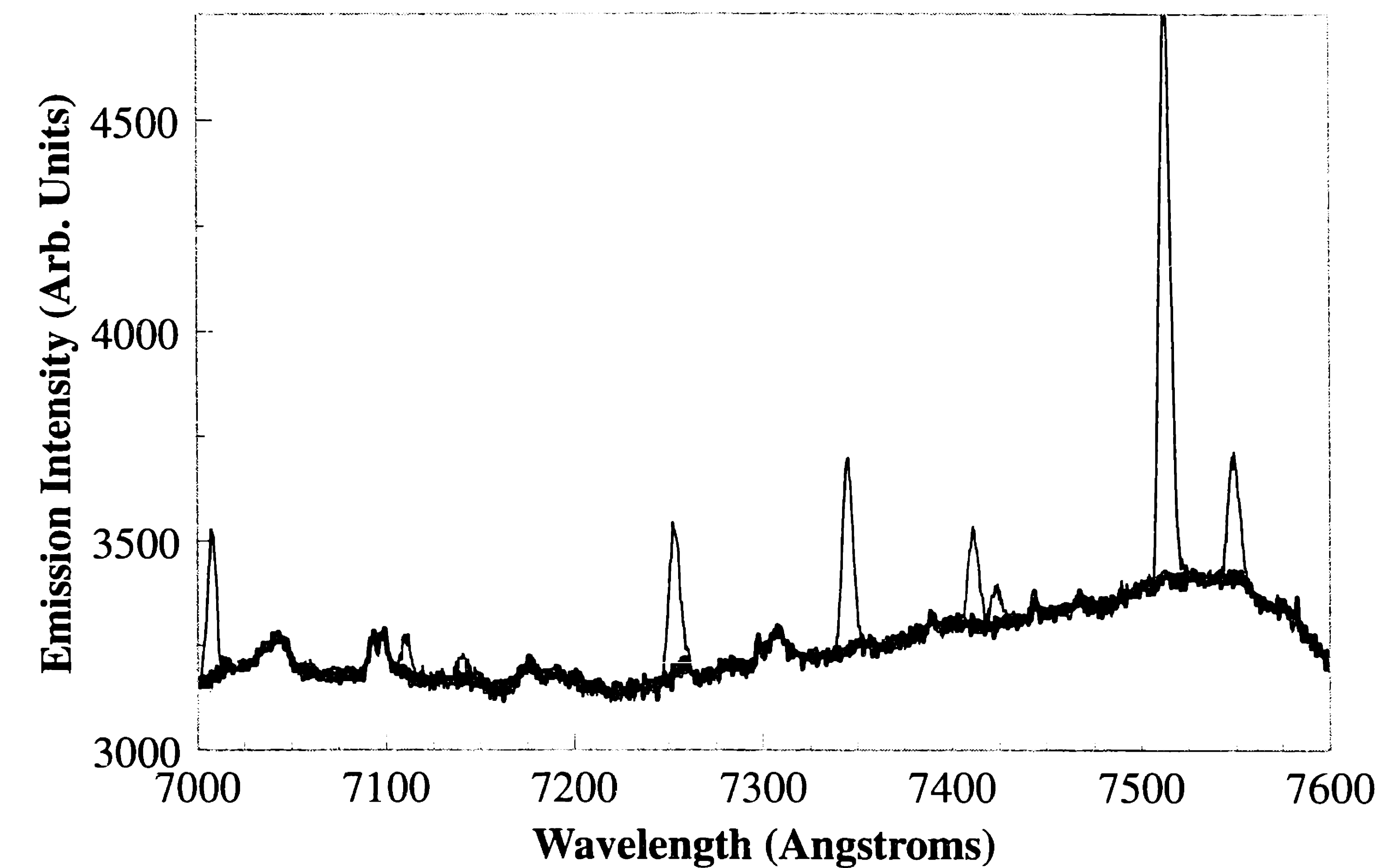


Figure 2-7: Plasma Optical Emission Spectrum for HBr/Cl_2 Plasma Taken with Jarrell Ash Monospec 270-mm Monochromator With Linear Diode Array Showing Background Spectrum (Black) and Measured Spectrum (Gray).

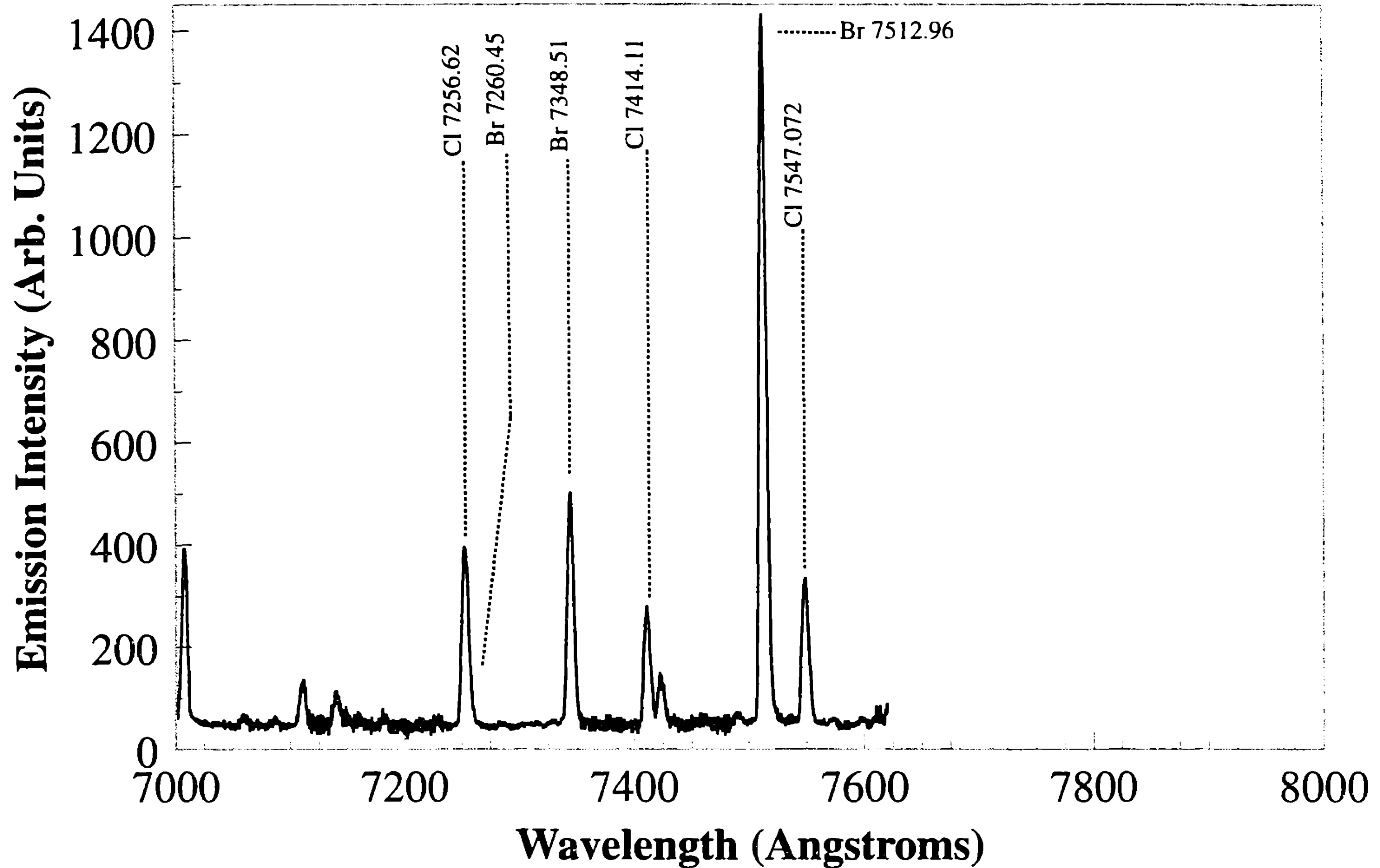


Figure 2-8: Plasma Optical Emission Spectrum for HBr/Cl_2 Plasma Taken with Jarrell Ash Monospec 270-mm Monochromator With Linear Diode Array.

C) Jobin Yvon

The third monochromator available on the AME-5000 was a 640-mm focal length f/7 JobinYvon (Instruments SA, Inc. (Edison, NJ)) HR640. Two different 1800 groove/mm diffraction gratings were installed on the HR640: grating "A" (4500 Å-10000 Å) and grating "B" (2500 Å Blaze (~ 1650 Å - 4500 Å)). The maximum resolution of the HR640 was approximately 0.09 Å. A Hamamatsu (Hamamatsu Corp. (Middlesex, NJ)) R955 photomultiplier tube was used as the detector. The PMT current was measured using a Keithly Instruments (Cleveland, OH) 485 autoranging picoammeter or a Keithly Instruments 427 current amplifier.

To control the monochromator, a SpectraLink (Instruments SA, Inc. (Edison, NJ)) was used. The modules contained within the SpectraLink were: PS (power supply for the unit), HV (high voltage for the PMT), MDR (motor drive for moving the diffraction grating to change wavelength), INT (interface for a computer), MCC (motor controller to change gratings or to change exits (axial/lateral)), and PIO (a second motor controlling board).

A spectrum taken with the HR640 for a HBr/Cl_2 plasma is shown in Figure 2-9 for the same plasma conditions as for Figures 2-6 and 2-8. A large improvement in resolution is seen when using the HR640. The resolutions for the different monochromators in terms of the full-width half-maximum (FWHM) for the Br 7512.96 Å peak are listed in Table 2-4.

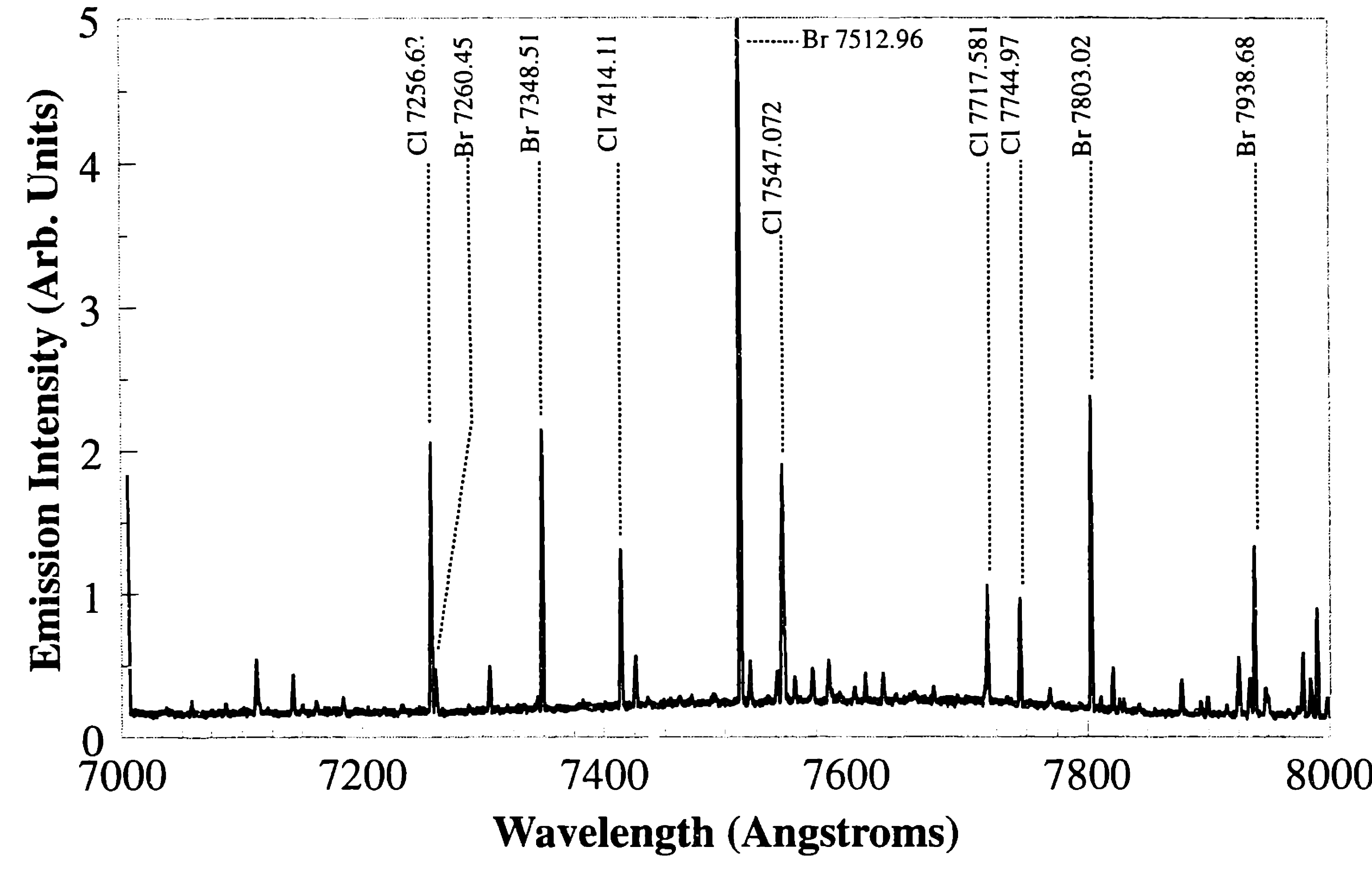


Figure 2-9: Plasma Optical Emission Spectrum for HBr/Cl_2 Plasma Taken with Jobin Yvone 640-mm Monochromator.

Table 2-4

Comparison of Three Different Monochromators

Monochromator	Focal Length (mm)	FWHM of 7512.96 Å Peak	Slit Size Used (μm)
Xinix	100	14.3	250
Jarrell-Ash	275	7.0	1000
JobinYvon	640	2.2	300

The resolution of the Xinix monochromator was not adjustable. The resolution of the Jarrell-Ash was adjustable by changing the inlet and outlet slits (removable fixed slits). The resolution of the JobinYvon was adjustable by varying the slit height and width using micrometers on the instrument.

2.2.3 RF Power Monitor

An easily controlled plasma parameter is the plasma power; the plasma power measured at the RF generator is an easily monitored and controlled value. However, it has been shown that the power actually dissipated in the plasma is not the same as the power at the RF generator, and in fact, the amount of power dissipated in the plasma may be only a small fraction of the input power, depending on the process conditions and chemistry [Butterbaugh *et al.*, 1990],[Butterbaugh and Sawin, 1992].

A Comdel real power monitor (Comdel RF Power Systems (Beverley, MA) #RPM-1) was used to monitor plasma electrical properties [Zau *et al.*, 1991]. The RPM

consisted of sensor head and a control unit. The sensor was mounted within the AME-5000 "water-box" such that it was in-line between the matching network and the feed-through to the cathode. The RPM measured the rms values of the RF voltage and current, and the phase angle between the two. From this, it determined the plasma impedance and the actual power dissipated in the plasma. Due to the lack of a DC current path in the AME-5000 polysilicon etch chamber, the RPM was unable to measure the DC bias. The control unit was rack mounted within the AME-5000 mainframe. This method is not invasive because it does not go inside the plasma. However, it is also not non-invasive due to the placement of the sensor after the matching network. It lies somewhere in between the two extremes of plasma diagnostics.

Sample RPM output signals are shown in Figures 2-10 and 2-11. The rms values of the RF voltage and RF current are plotted in Figure 2-10. For this case, the plasma was ignited, and the matching network was held at a fixed value after tuning, to accentuate the changes visible at end-point. The current and voltage shown an initial overshoot during ignition, settling down to steady-state values. At about 70 seconds, both signals shown a rapid change, due to the change in the plasma electrical characteristics as a polysilicon film on the wafer is removed and a silicon dioxide film is exposed. The phase angle between the current and voltage waveforms (Figure 2-11) is stable and does not shown a significant change at end-point. The phase angle is very close to -90°, indicating a very capacitive plasma. In some cases, the phase angle was too close to -90° to permit the measurement of power due to the limitations of the RPM-1. The plasma power (Figure 2-11) shows a drop of about 8 W at endpoint, due to the decreased current and voltage.

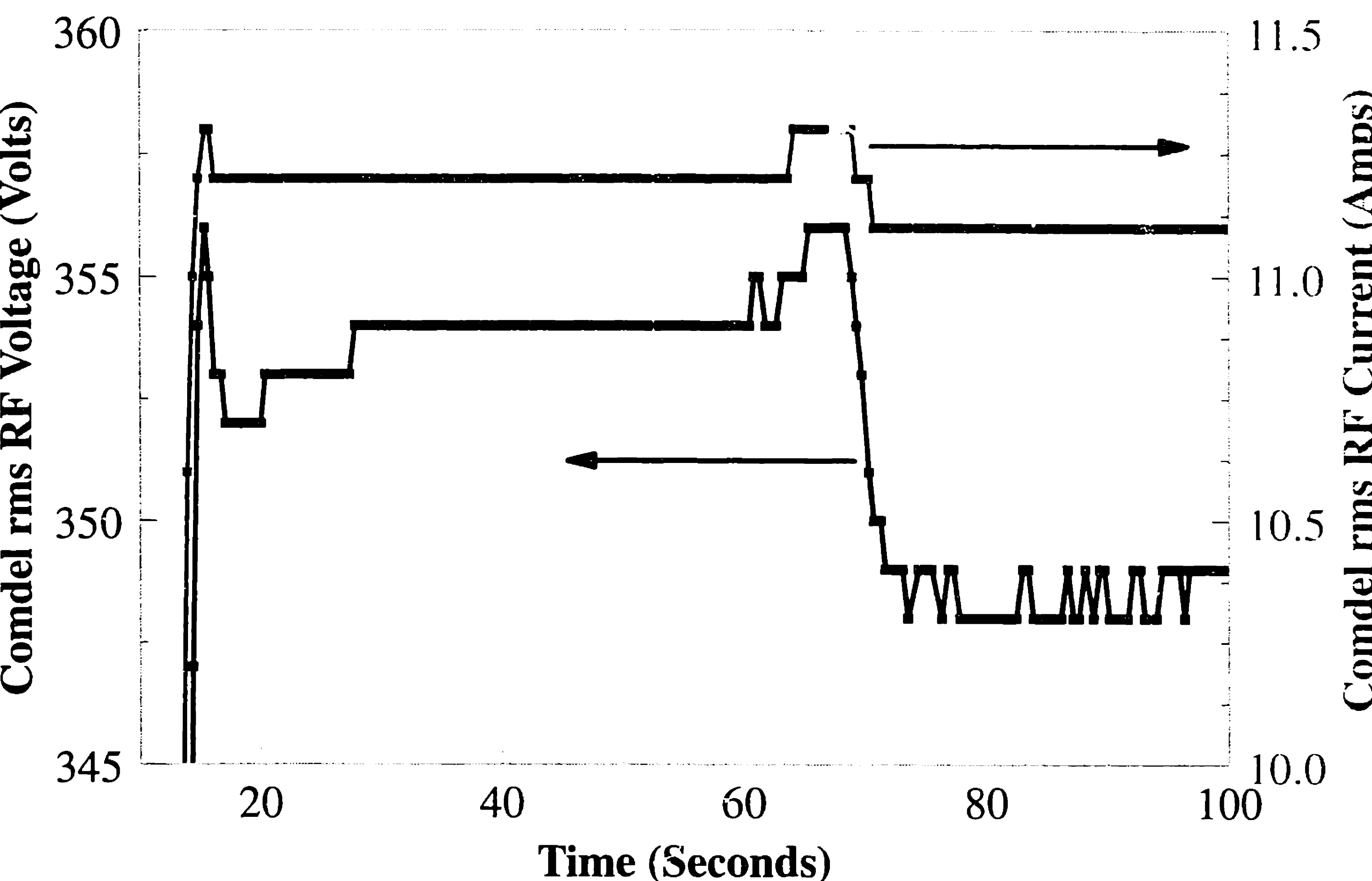


Figure 2-10: RMS values of Plasma RF Voltage and RF Current Measured using a Comdel RPM-1. Signal Variation at 70 Sec. is Due to Polysilicon End-Point.

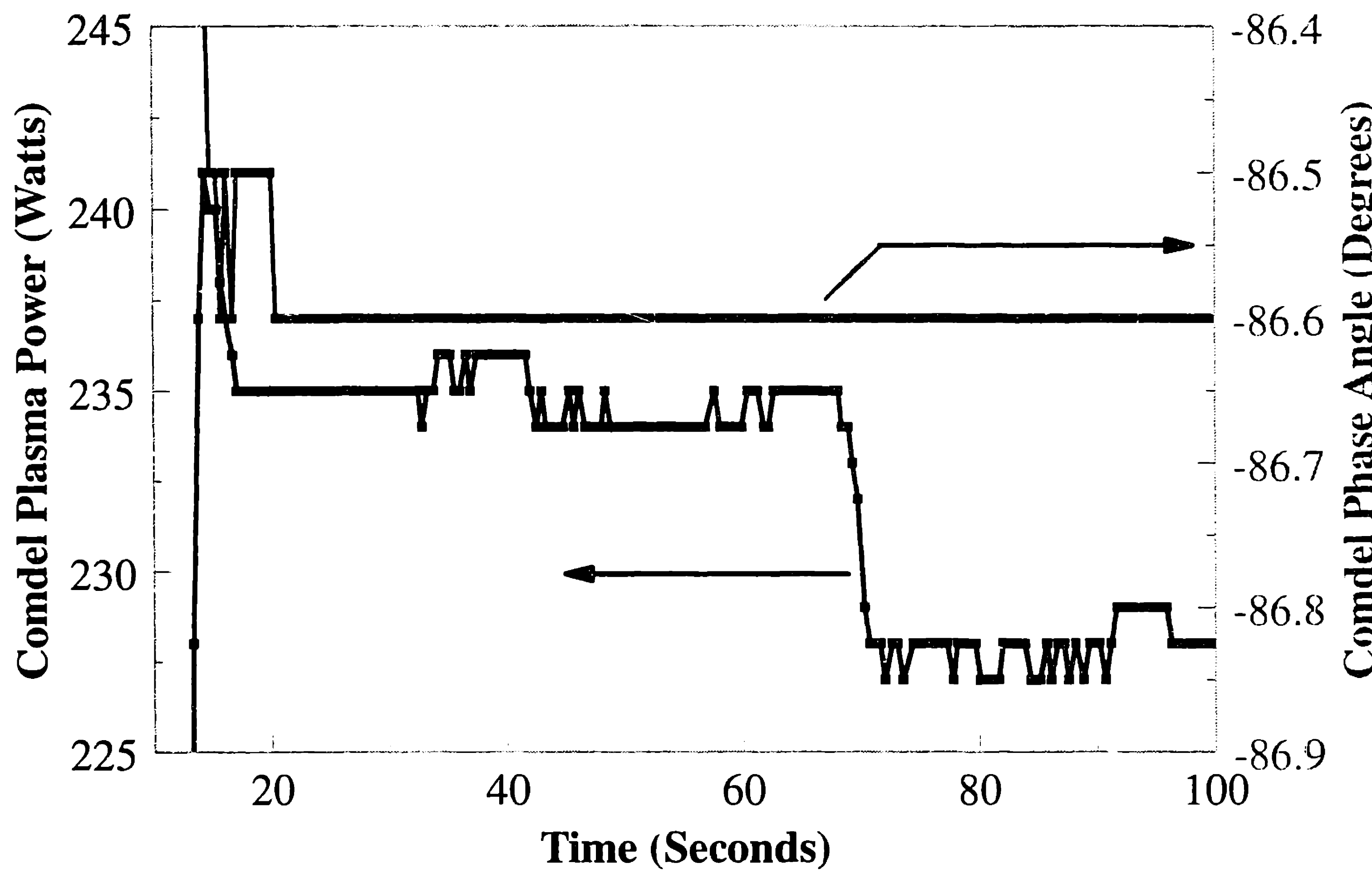


Figure 2-11: Phase Angle Between RF Current and Voltage Waveforms, and Calculated Plasma Power, Measured Using Comdel RPM-1. Signal Variation at 70 Sec. is Due to Polysilicon End-Point.

The voltage measured with the RPM-1 was compared to the voltage measured on a Lecroy 9400 oscilloscope using a Tektronix 100x voltage probe attached to the cathode backside (Figure 2-12). There is a linear relationship between the two voltages, with the voltage measured with the Lecroy being about 87.6% of the voltage measured with the Comdel. The difference may be due to the use of an uncalibrated 100x voltage probe.

Plasma electrical properties were measured over a wide range of plasma parameters: power: 125-375 watts, pressure: 20-180 mTorr, magnetic field: 0-150 Gauss, HBr flowrate: 0-20 SCCM, total flow rate HBr and Cl_2: 40 SCCM (see Chapter 3 for more details). The plasma power measured with the RPM-1 is plotted as a function of the plasma power measured at the RF generator (Figure 2-13); a total of 39 experiments were conducted. Over this wide range of conditions, a linear relationship between RPM-1 power and generator power was observed, fitting the form of : $P_{comdel} = 0.761 P_{generator} - 18.9$ watts. This indicates that approximately 76 percent of the plasma power created at the RF generator is passing through the RF matching network and into the plasma. Thus, the loses in the matching network and in the cable to the generator are on the order of 25% of the power. Only the RPM-1 power correlated with the generator power; the current, voltage and impedance did not. The correlation coefficients (R^2) for linear fits between these electrical properties and the nominal generator power are listed in Table 2-5.

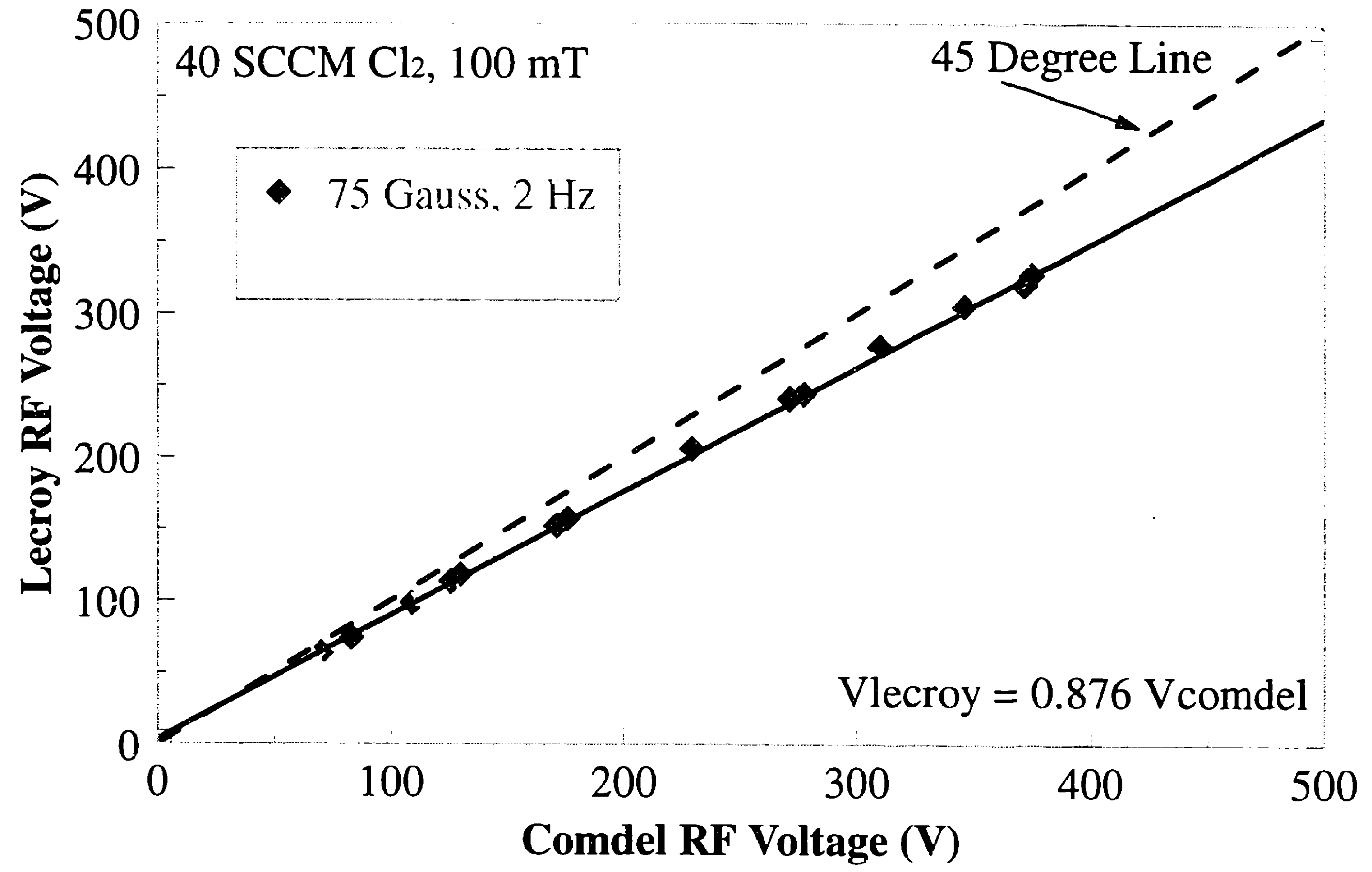


Figure 2-12: Comparison of Comdel RPM Voltage With Tektronix 100x Voltage Probe and Lecroy 9400 Oscilliscope.

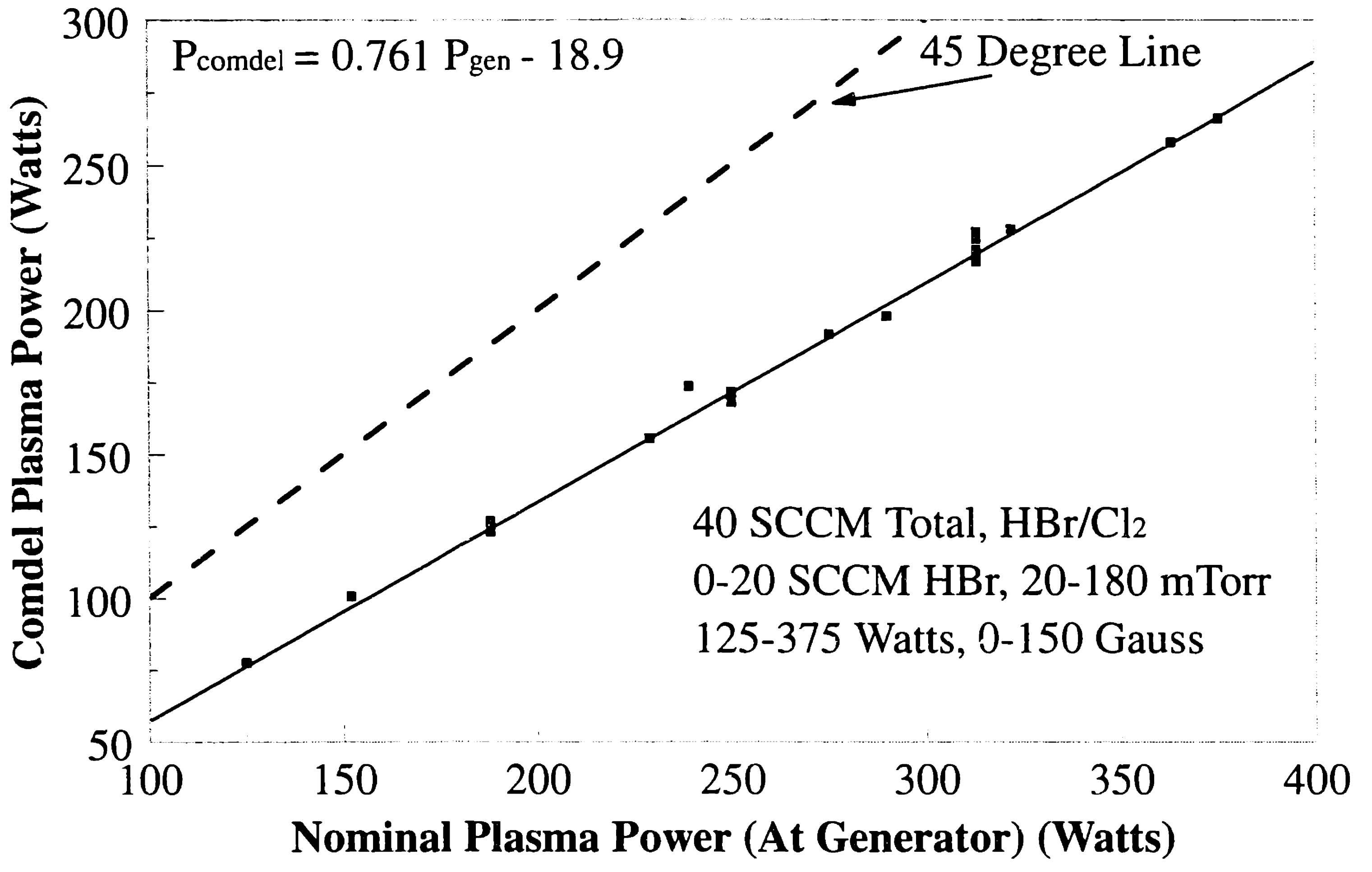


Figure 2-13: Comparison of Comdel RPM Plasma Power and Nominal Plasma Power Measured at the RF Generator for 39 Experiments Over a Wide Range of Experimental Conditions.

Table 2-5

Correlation Coefficients for Different Electrical Parameters

Variable	R^2
Power	0.996
Voltage	0.579
Current	0.668
Impedance	0.0245

2.2.4 Wafer and Cathode Temperature

During plasma etching, the wafer temperature rises due to energetic ion bombardment of the wafer surface and to exothermic etching reactions occurring on the wafer surface. Traditional temperature measuring methods such as thermocouples will not work in a plasma environment due to the RF electrical field and the rotating magnetic field. An optical temperature measuring method was used. Because of the placement of probes within the etching chamber, this was an invasive diagnostic.

Both the wafer temperature and the cathode pedestal temperature were measured using a Luxtron (Luxtron Corp. (Santa Clara, CA)) 755 Fluoroptic® thermometer. This technique has been proven to be effective at measuring wafer temperature during etching [Egerton, 1982]. The Luxtron 755 consists of a control unit and fiber-optic temperature probes. The probes have a temperature sensitive phosphor sensor at the tip of the fiber. The control unit provides a xenon strobe lamp to excite the phosphor and determines the temperature from the measured decay time of fluorescent radiation from the phosphor.

This optical temperature measurement technique is superior to thermocouples because it does not suffer from RF interference.

Two MIW type probes (-190°C to +300°C operational range) were inserted into the etching chamber via the helium backside cooling line. One probe touched the cathode pedestal upon which the wafer was clamped during etching, directly below the center of the wafer, and the other probe touched the center of the wafer backside (Figure 2-14). Alternatively, a probe was placed on the RF feedthrough Cu post on the back of the cathode. However, the temperature here was very stable due to the large cooled thermal mass between the plasma and the point of measurement.

Good thermal contact was ensured because the probes were surrounded by up to 10 Torr of He during etching. Cathode temperature was controlled by an ultra low temperature recirculating cooler (FTS Systems Inc. (Stone Ridge, NY) #RC-210C-21) instead of the standard Neslab chiller to allow for lower cathode temperatures. The use of the Luxtron to characterize thermal processes occurring with in the AME-5000 is described in much greater detail in Chapter 6.

2.2.5 Full Wafer Interferometry

As mentioned in Section 2.2.1, etching rates determined with LI were not spatial resolved; only a single laser beam and photodiode were used. We desired to measure the etching rate uniformity across a wafer during etching. The first method used (described in Chapter 3) was to measure film thickness at a number of sites before etching, partially etch the film, and them measure the film thickness again. However, this is a slow and

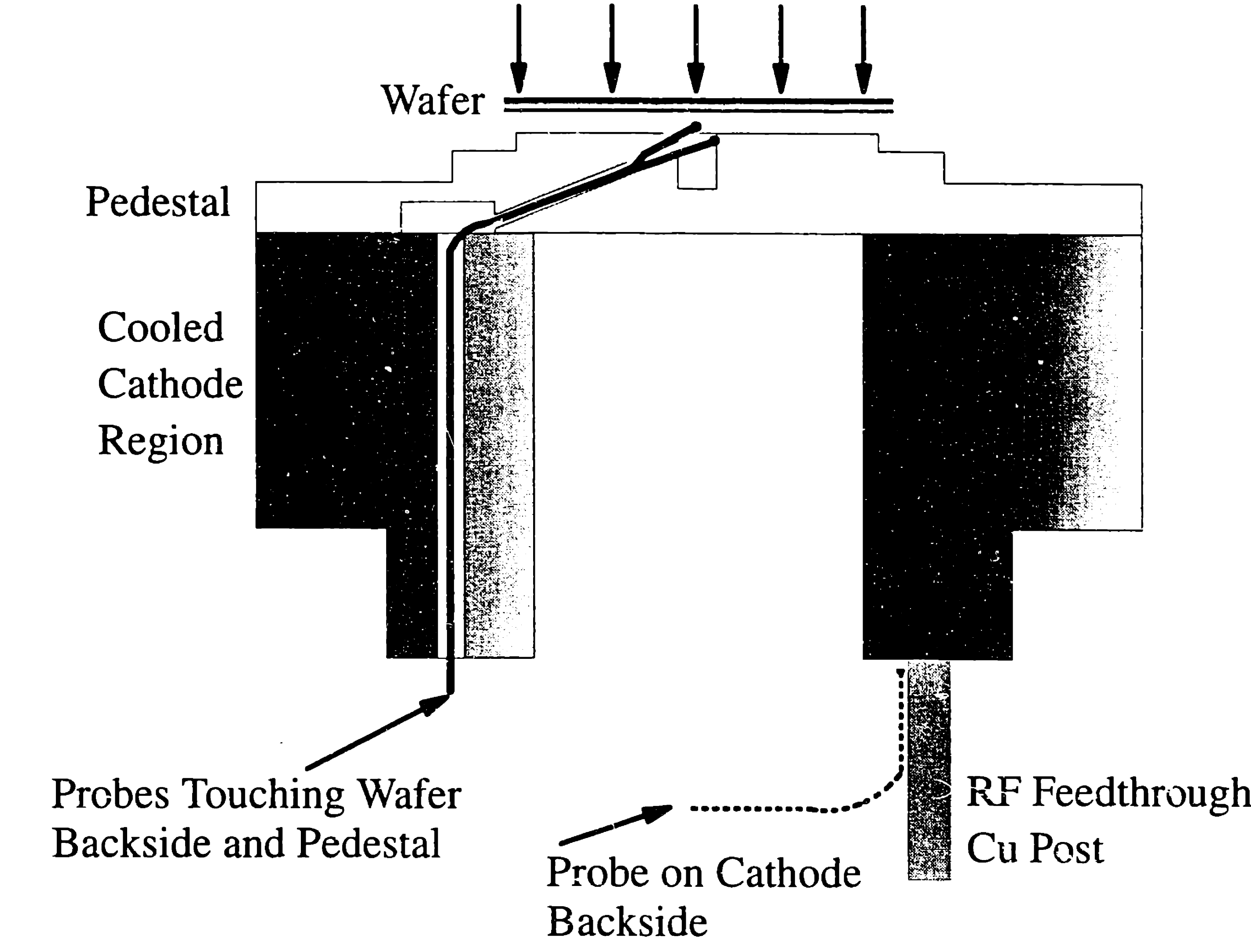


Figure 2-14: Location of Luxtron Temeperature Probes on RF Cathode.

inconvenient method. We considered splitting a laser beam and using a series on photodiode detectors, but did not pursue this idea. Instead, we invented a new method for measuring etching rates across an entire wafer during the plasma etching process.

The full wafer interferometry (FWI) system used an Electrim EDC-1000HR 8-bit Charge Coupled Device (CCD) camera; a 16-mm focal length f/1.6 lens was used to completely image a 100-mm wafer through a 50-mm diameter optical viewport during etching. Interferometry signals were observed using an optical bandpass filter at either 7534 Å for optical emission interferometry (OEI) or at 6328 Å for LI. This invention is described in much greater detail in Chapter 5.

2.3 Data Acquisition

All of the plasma diagnostics mentioned in Section 2.2 were interfaced to a computer to control data acquisition. An IBM-compatible 80286 personal computer (PC) was used to control the Luxtron temperature monitor, the Comdel power monitor, the Jarrell-Ash monochromator, and the JobinYvon monochromator. All of these diagnostics could be run simultaneously. Optionally, the 80286 could be used to control the Xinix monochromators, both to scan to plasma OE and as an endpoint control for the AME-5000. A 33-MHz 80386 PC was used to control the FWI system. Data collection on the 80386 could be started either by the user, or automatically by the 80286 PC to synchronize acquisition with the 80286.

2.3.1 Data Acquisition Hardware

The 80286 was equipped with a general purpose interface card made by Data Translation (Marlboro, MA); the 12-bit DT2811-PGH contained: 8 differential analog to digital (A/D) inputs with programmable gains of 1,2,4 and 8; 2 digital to analog (D/A) outputs; 8 digital inputs (DI); and 8 digital outputs (DO). The board was configured to accept 0-5V input signals. A Data Translation DT2817 (32 bit digital input/out (DIO)) board was used to control the TracorNorthern TN6200 DARSS.

The Output of the LI photodiode was fed to A/D #0. A 1/2 voltage divider using 10 KΩ resistors was used to bring the 0-10V signal into the required 0-5V range. The output from the Keithley 427 current amplifier used with the PMT on the HR640 was fed to A/D #1. DO #0 was used to synchronize data collection with the 80386. A TTL <FALSE> was the default state. The bit switched to TTL <TRUE> when data collection was to begin, and back to TTL <FALSE> when data collection was to end.

An IOtech (Cleveland, OH) Personal488 GPIB/IEEE488 interface board was also installed. This board was used to control the Luxtron temperature monitor and the Keithley 485 picoammeter. Two serial ports on the 80286 were also used. One was used to control the SpectraLink for the HR640 monochromator; the other was used to interface with the control unit for the Comdel RPM-1.

The 80386 PC used a digitizer board that was part of the Electrim CCD to acquire CCD images. A Data Translation DT2817 DIO board was also used in this PC for several purposes: (1) to receive the signal for the start and stop of data collection when synchronized to the 80286 PC; (2) to power the Hall effect probe for magnetic field rotation synchronization (see Chapter 5); and (3) to read in the signal from the Hall effect probe.

2.3.2 Data Acquisition Software

A variety of software was written for data acquisition. The software for the 80286 to control everything except the CCD was written in Turbo PASCAL 6.0 (Borland Intl. Inc. (Scotts Valley, CA)). The software for the 80386 to control the CCD was written in Microsoft (Redmond, WA) C 7.0. Printouts of all of the software is included in the appendices.

2.4 Optical Emission Characterization of Plasmas

A study of plasma optical emission was undertaken to identify emission lines for use in actinometry (explained in Section 2.4.3). The use of actinometry for plasma characterization on the AME-5000 will discussed in Chapter 3.

2.4.1 Long Range, High Resolution Plasma Scan - HR640

As mention above, the HR640 provided the highest resolution of all 3 monochromators used. A large range scan of plasma optical emission was done using the HR640 to provide a reference library for later use in searching for optical emission lines for identification of species present in the plasma (see Section 8.4.1). Such large range scans were not done frequently because of the very long time required to use the HR640 as a scanning monochromator.

Scans were made for a 100 mTorr plasma with 30 SCCM Cl_2 and 10 SCCM HBr

from the ultraviolet (UV) (1750 Å) to the infrared (IR) (10,000 Å). Figure 2-15 shows a scan from 1750 Å to 4750 Å using grating B and Figure 2-16 shows a scan from 4250 Å to 8250 Å using grating A. Optical emission in the deep UV is limited by strong absorption by O_2 below 1900 Å [Pearse and Gaydon, 1984]. These two figures shown the optical emission spectrum of this plasma is quite complex, with many emission lines and continua. The origin of some of these features is discussed below.

2.4.2 Xinix and DARSS Scans

The presence of atomic, ionic, and molecular emission spectra for several species makes the interpretation of Figures 2-15 and 2-16 very complex. In order to understand the OE spectra shown, spectra were recorded for the individual components present in the plasma using both the Xinix and Jarrell-Ash monochromators.

A) HBr Plasmas

The optical emission spectrum for a 100 mTorr HBr plasma taken with the Xinix monochromator is shown in Figure 2-17. This spectrum is also quite complex due to emission from such species as Br, Br^+, Br_2, Br_2^+, HBr, HBr^+, H, and H_2. Some bands can be identified, such as the strong band just below 3000 Å (and the strong band just above 3500 Å which are caused by Br_2 emission [Pearse and Gaydon, 1984].

Emission lines separate from continua (*e.g.*, the strong continuum near 4000 Å)

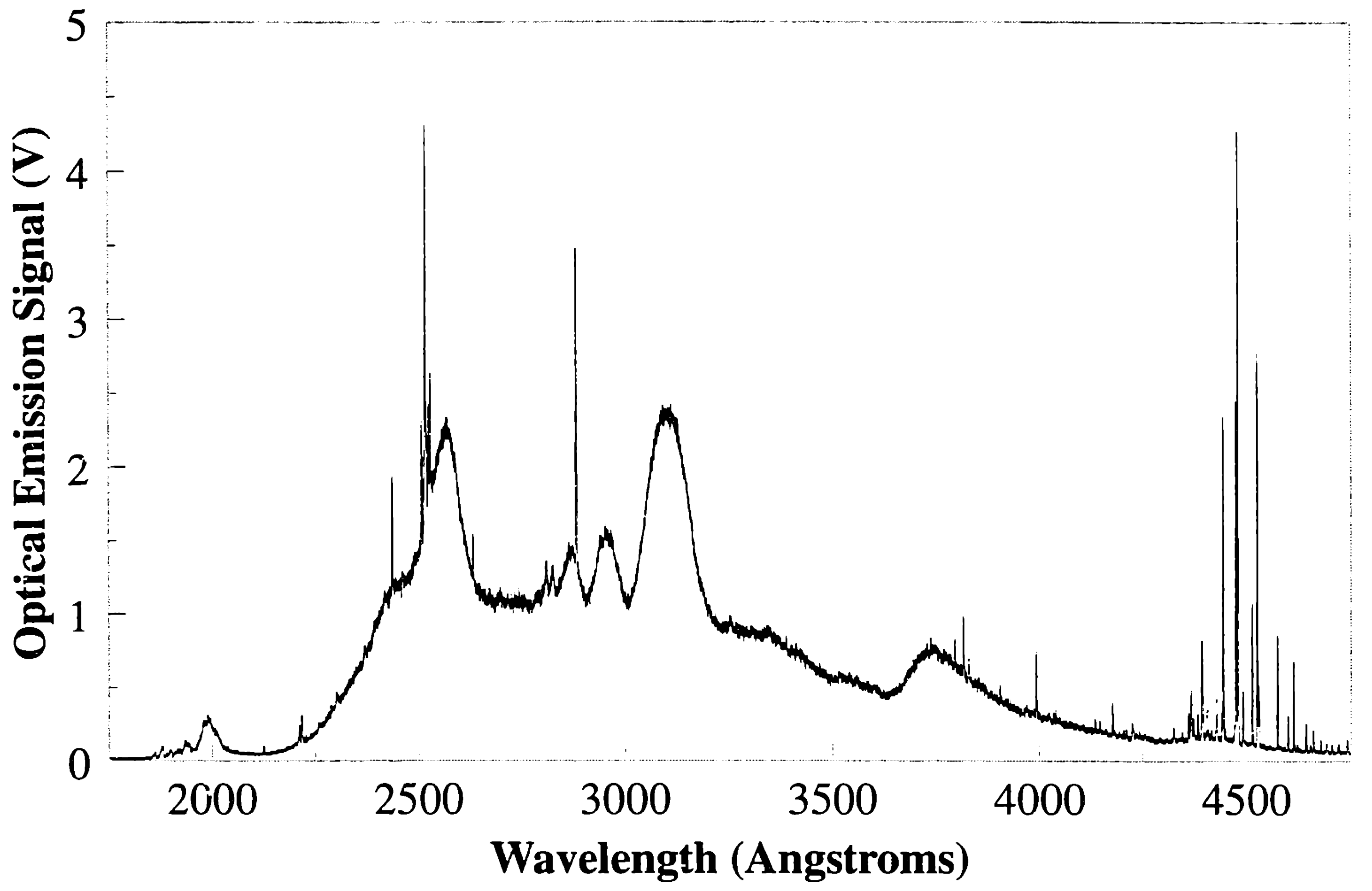


Figure 2-15: Plasma Optical Emission Spectrum for 100 mTorr HBr (10 SCCM) / Cl_2 (30 SCCM) Plasma taken With Grating "B" on HR640 Monochromator

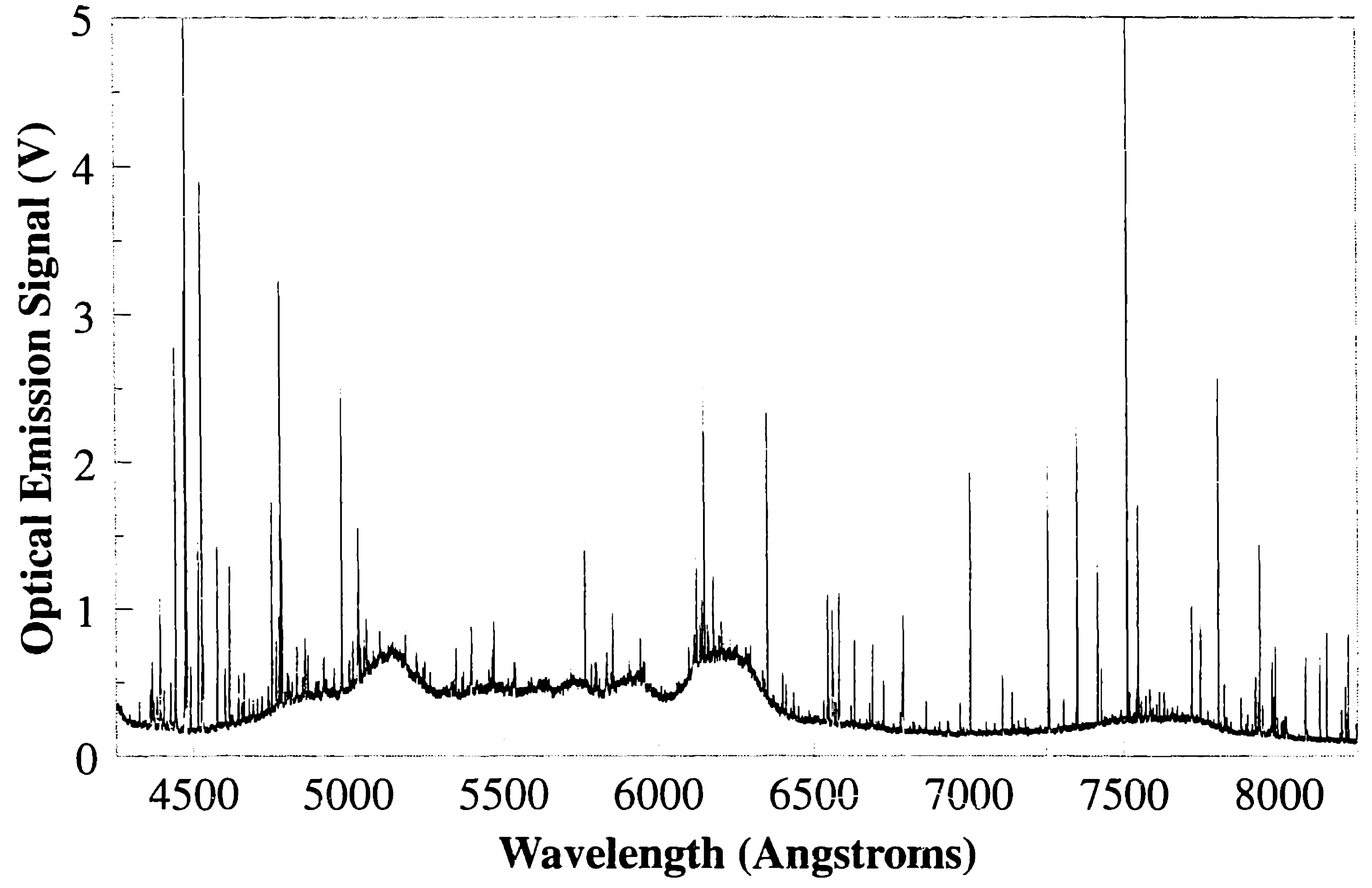


Figure 2-16: Plasma Optical Emission Spectrum for 100 mTorr HBr (10 SCCM) / Cl_2 (30 SCCM) Plasma taken With Grating "A" on HR640 Monochromator

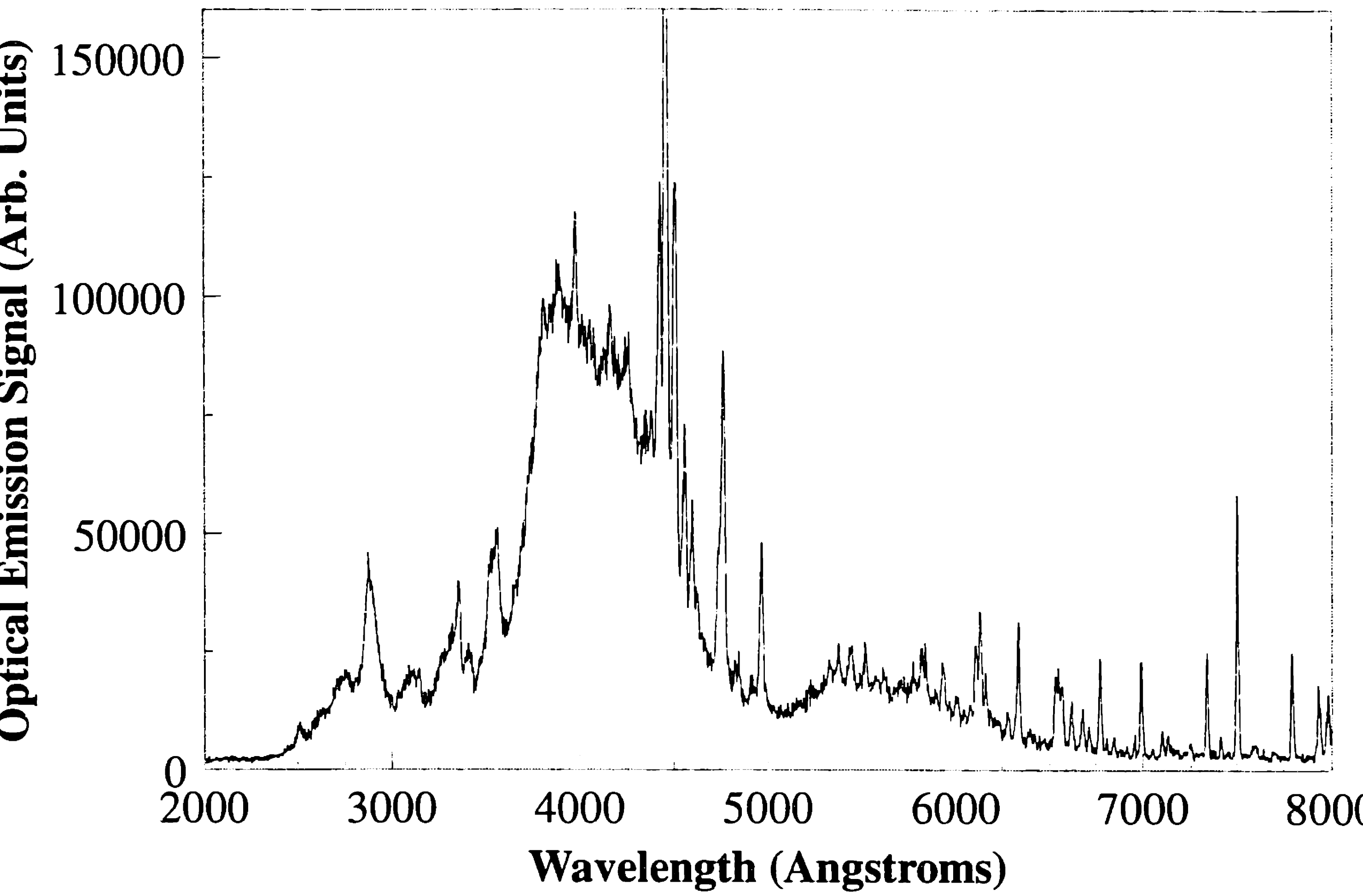


Figure 2-17: Xinix Plasma Optical Emission Spectrum for 100 mTorr HBr

are the easiest to identify. More detailed plasma spectra from 8000 Å to 8600 Å and from 6950 Å to 7575 Å taken with the Jarrell-Ash monochromator and the DARSS are shown in Figures 2-18 and 2-19, along with the assignment of a number of peaks. The information from these figures will be used is Section 2.4.3 for actinometry. Strong peaks are see for neutral excited Br at 8272.44, 8446.55, and 7512.96 Å.

Using the HR640 monochromator, we were able to identify emissions from atomic hydrogen: $H\alpha$ at 6552.852 Å and $H\beta$ at 4861.33 Å, but not $H\gamma$ at 4340.47 Å. Previous researchers were unable to identify atomic hydrogen emissions in HBr plasmas [Nakamura *et al.*, 1988][Bestwick and Oehrlein, 1990a]. Their failure resulted from the inability to resolve the $H\alpha$ line from the nearby, strong Br emission at 6559.80 Å using a lower resolution monochromator. Neither the Jarrell-Ash nor the Xinix monochromators was able to resolve the $H\alpha$ line.

B) Cl_2 Plasmas

The optical emission spectrum for a 100 mTorr Cl_2 plasma taken with the Xinix monochromator is shown in Figure 2-20 (compare to Figure 2-17). This spectrum is also quite complex due to emission from such species as Cl, Cl+, Cl_2, and Cl_2^+. The strongest continua from Cl_2 emission are at 2570 and 3070 Å, with a large number of emission bands for Cl2+ from 3862.6 to 5080.8 Å [Pearse and Gaydon, 1984].

Analogous to Figures 2-18 and 2-19, detailed plasma spectra from 8000 Å to 8600 Å and from 6950 Å to 7575 Å taken with the Jarrell-Ash monochromator and the DARSS are shown in Figures 2-21 and 2-22, along with the assignment of a number of peaks.

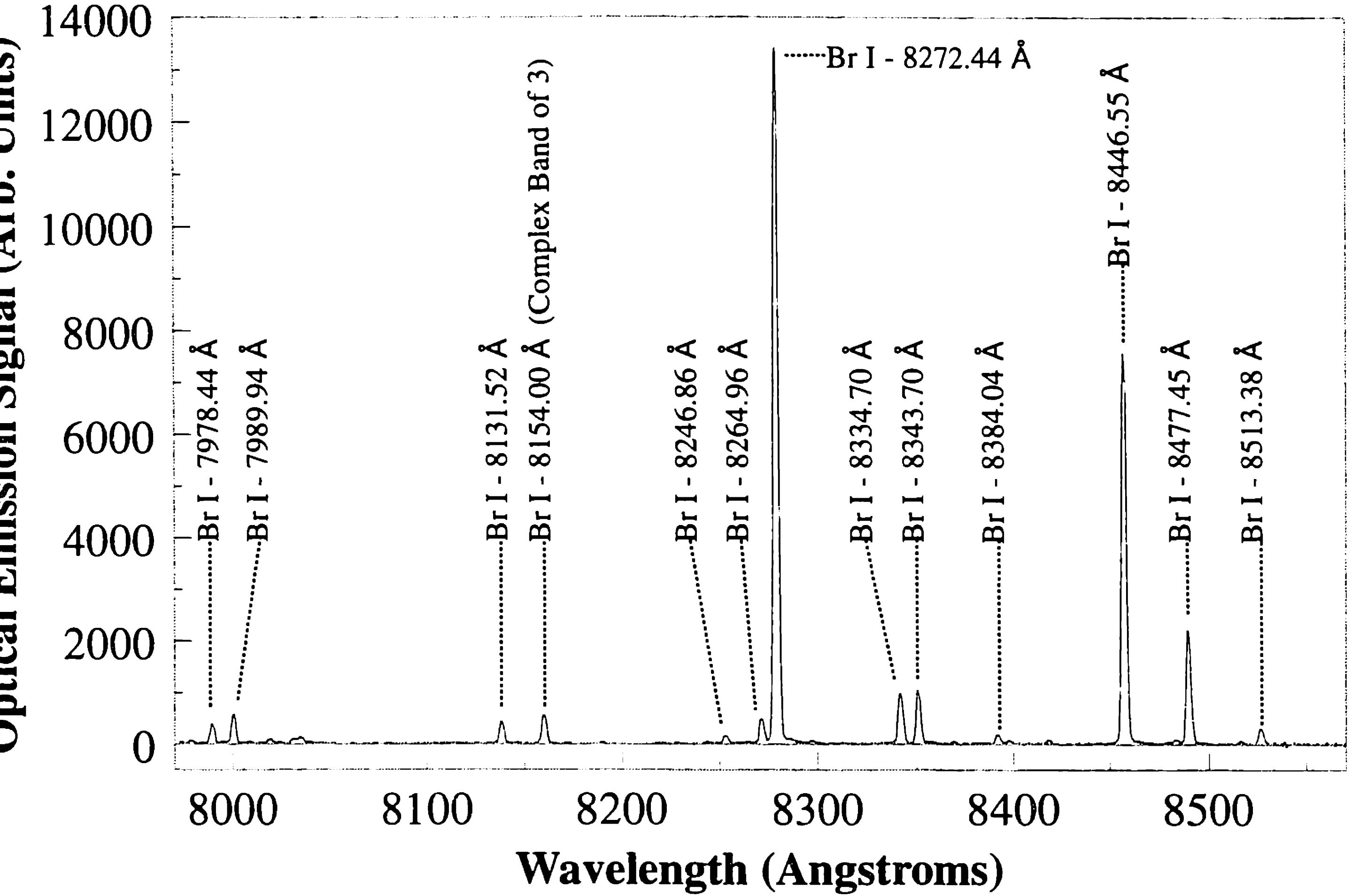


Figure 2-18: DARSS Plasma Optical Emission Spectrum for 100 mTorr HBr

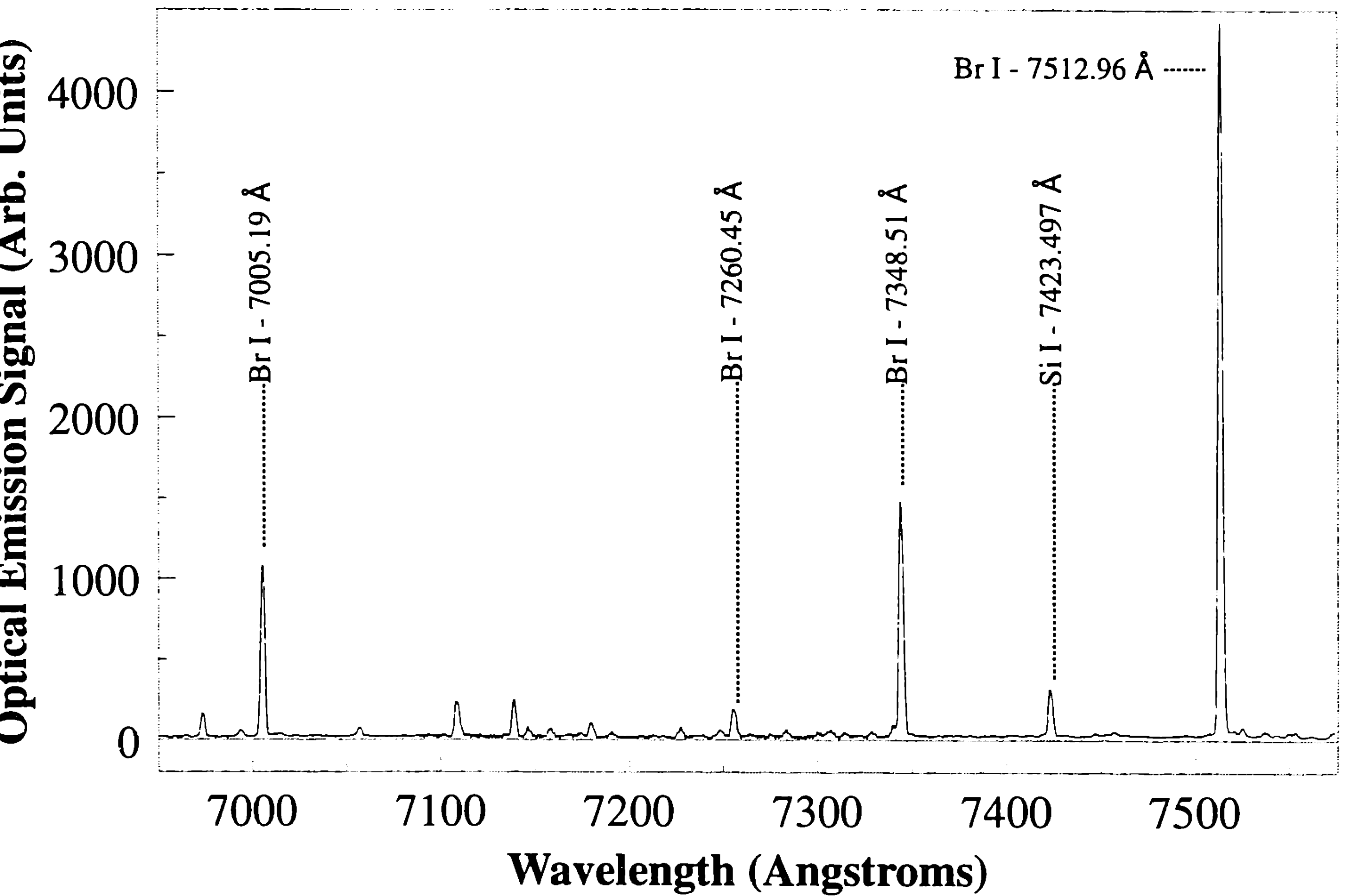


Figure 2-19: DARSS Plasma Optical Emission Spectrum for 100 mTorr HBr

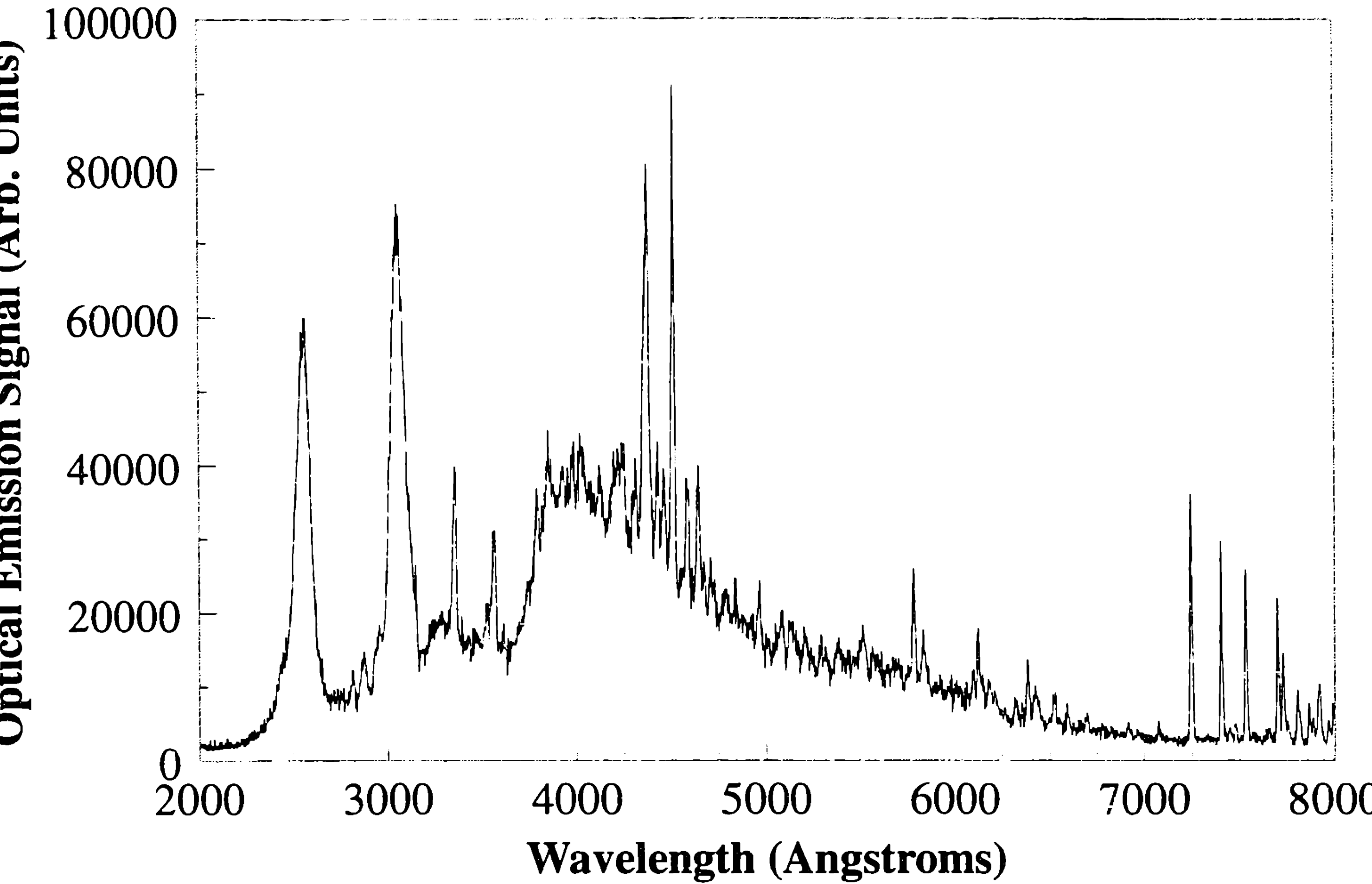


Figure 2-20: Xinix Plasma Optical Emission Spectrum for 100 mTorr Cl_2

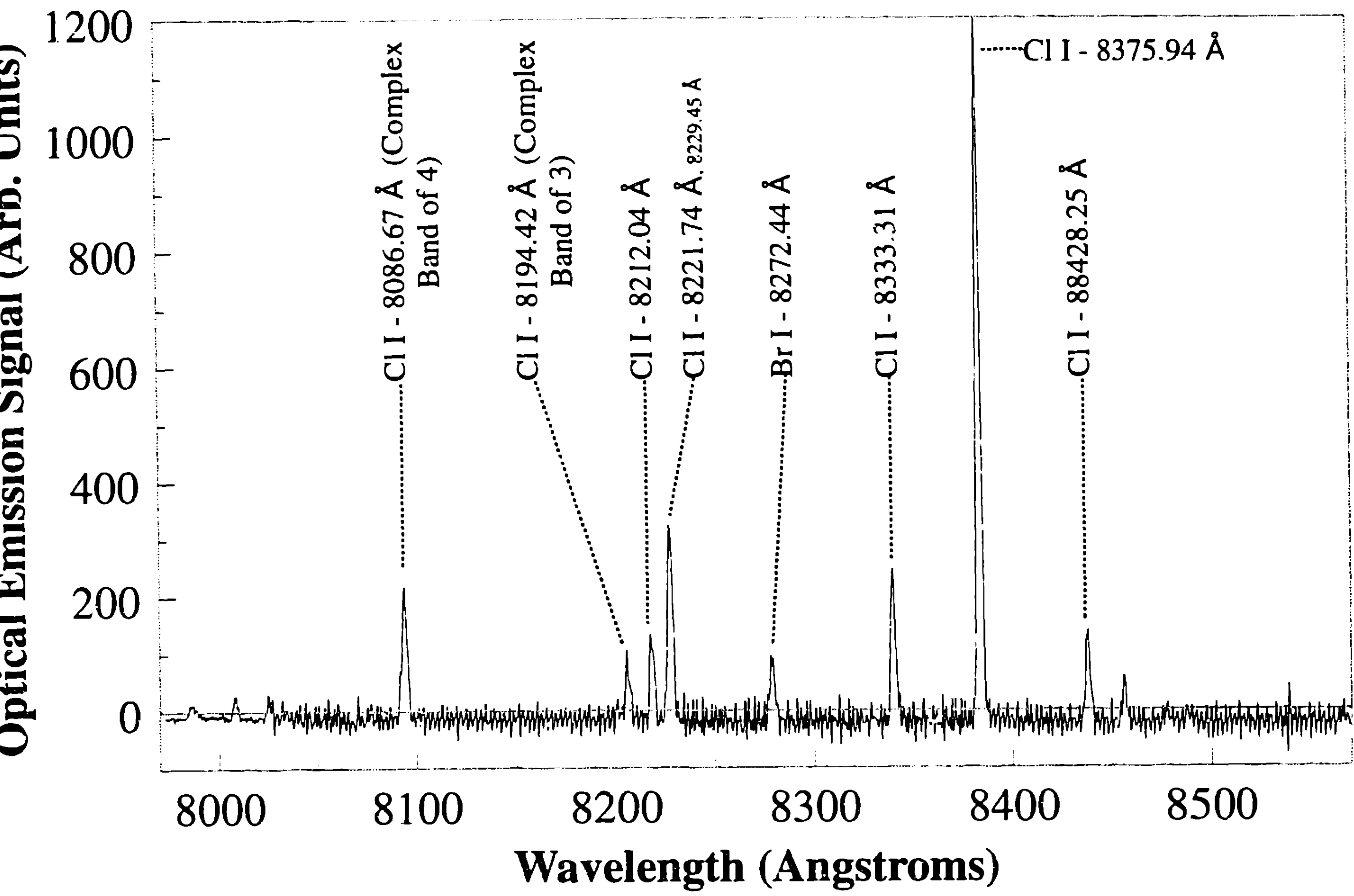


Figure 2-21: DARSS Plasma Optical Emission Spectrum for 100 mTorr Cl_2

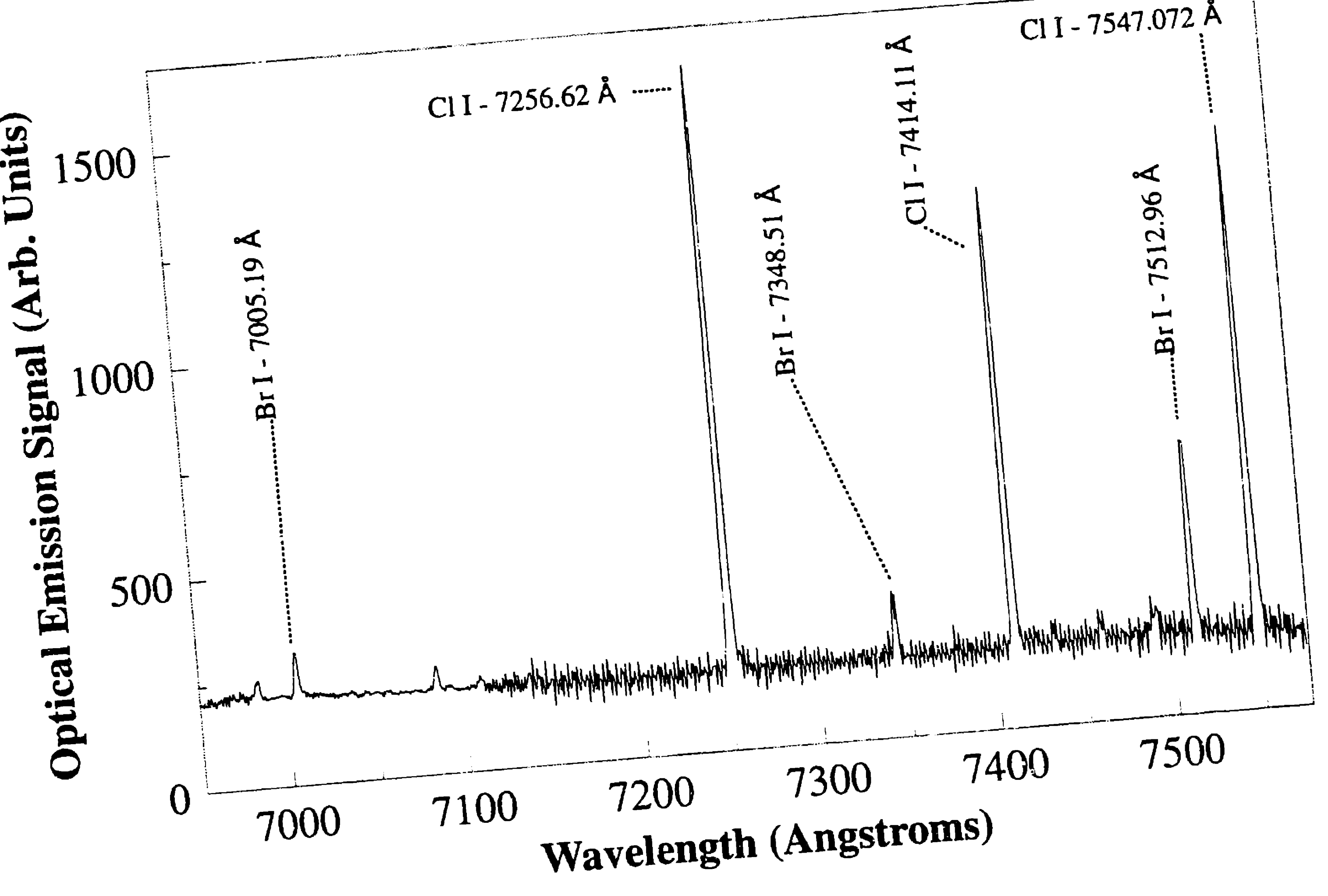


Figure 2-22: DARSS Plasma Optical Emission Spectrum for 100 mTorr Cl_2

Besides the strong emissions from Cl (8375.94, 7547.072, and 7256.62 Å), a number of Br emission lines are visible in the two spectra (7005.19, 7348.51, 7512.96, 8272.44 and Å). These lines are caused by Br desorbing from the chamber during processing and not from gas contamination. The Br peaks shrink and the Cl peaks grow during a Cl_2 plasma. Figure 2-23 is a plot of the peak areas for two Br emission lines (7005.19 and 7512.96 Å) and two Cl emission lines (7414.11 and 7547.072 Å) measured in a 100 mTorr Cl_2 plasma following a 100 mTorr HBr plasma. The two Br peak areas decrease with time, while the Cl peak areas slowly increase with time.

C) Ar Plasmas

The optical emission spectrum for a 100 mTorr Ar plasma taken with the Xinix monochromator is shown in Figure 2-24 (compare to Figures 2-17, and 2-20). The significance of Ar will be discussed in Section 2.4.3.

Detailed plasma spectra from 8000 Å to 8600 Å and from 6950 Å to 7575 Å taken with the Jarrell-Ash monochromator and the DARSS are shown in Figures 2-25 and 2-26, along with the assignment of a number of peaks. Very strong Ar emission lines are visible at 8115.311, 7514.652, and 7503.869 Å.

D) Combination Plasmas

The optical emission spectrum for a 100 mTorr, 30 SCCM Cl_2 - 10 SCCM HBr plasma taken with the Xinix monochromator is shown in Figure 2-27 (compare to Figures

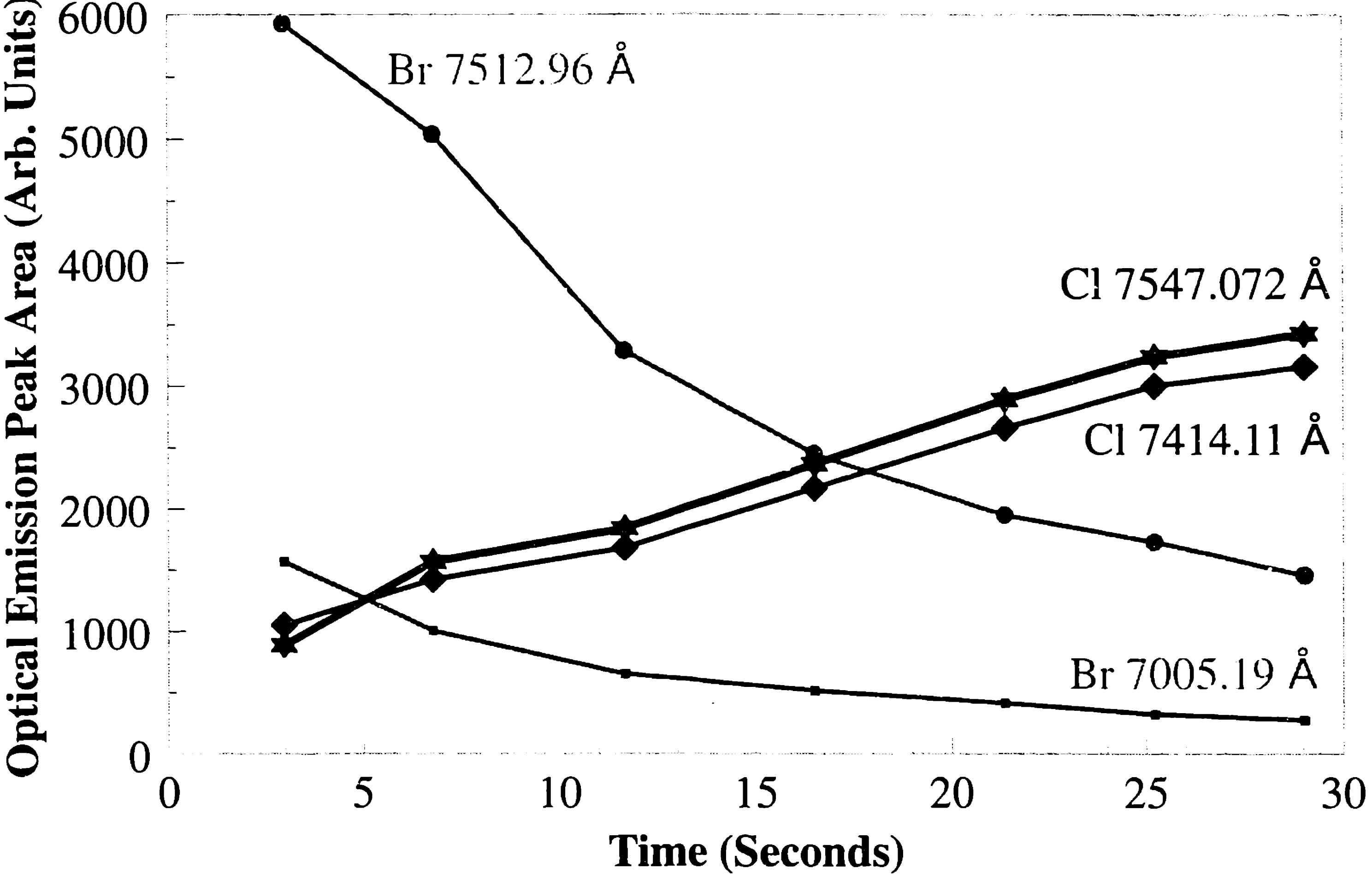


Figure 2-23: Evolution of Br and Cl Signals Over Time in a Cl_2 Plasma following an HBr Plasma.

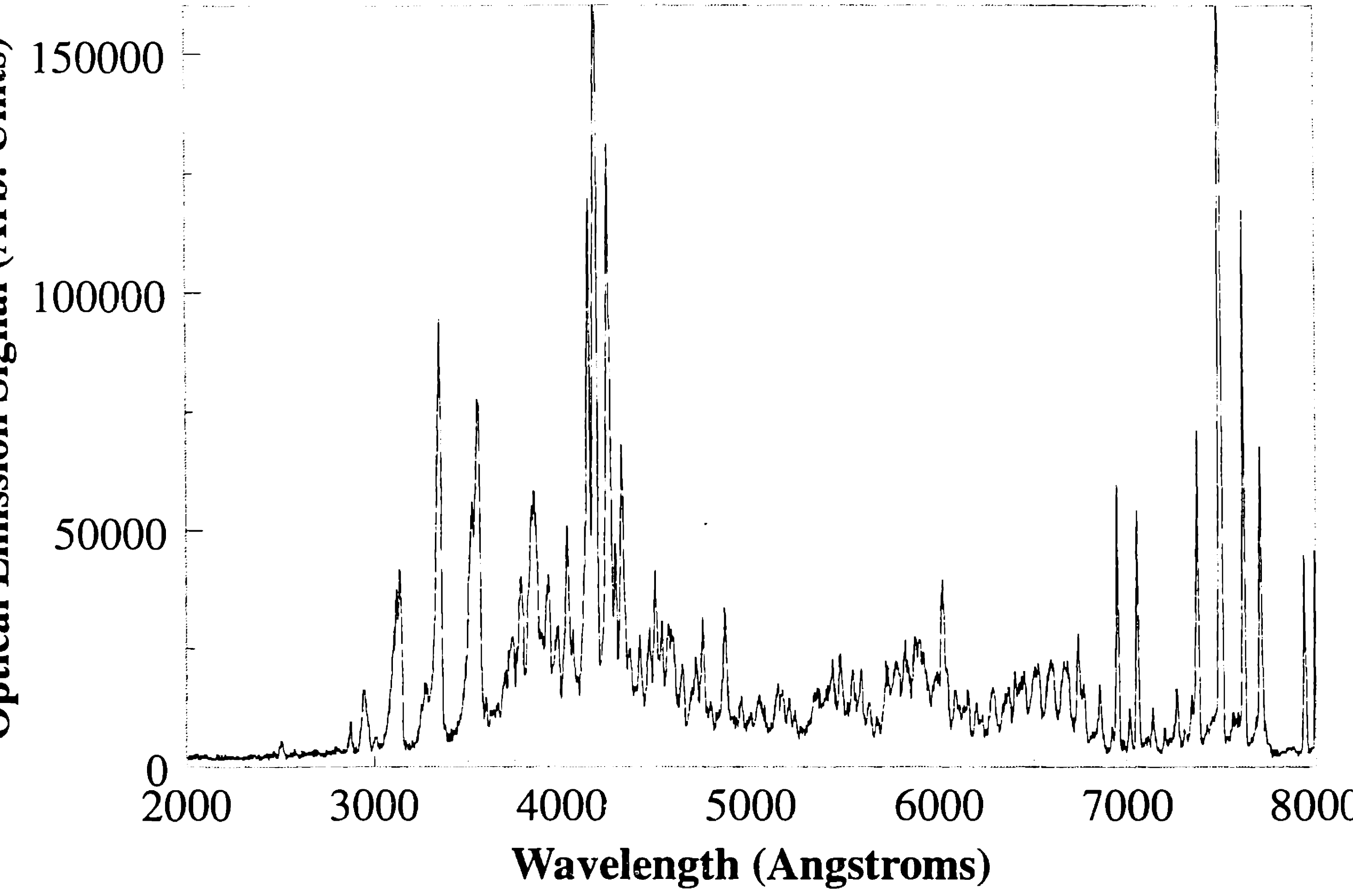


Figure 2-24: Xinix Plasma Optical Emission Spectrum for 100 mTorr Ar

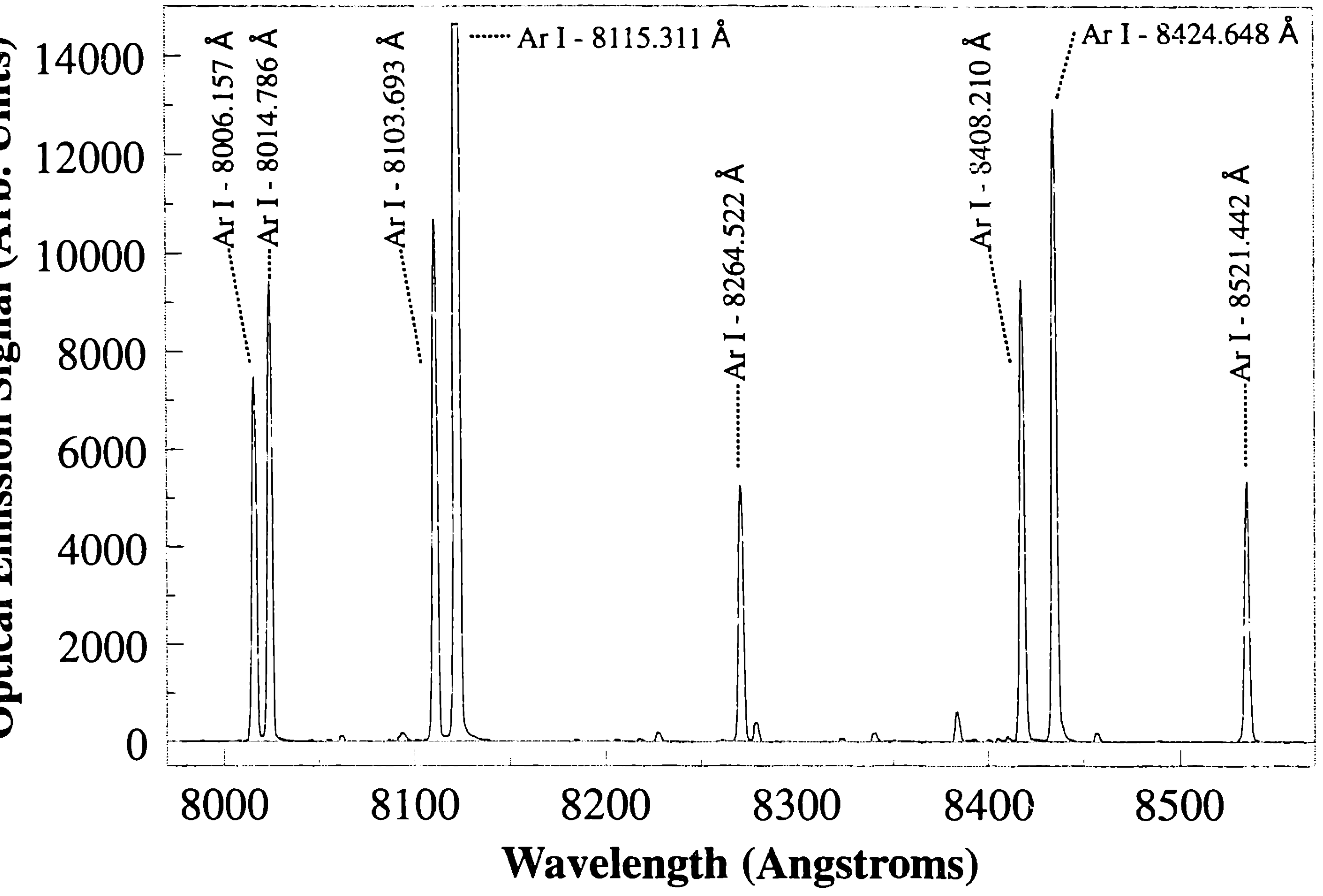


Figure 2-25: DARSS Plasma Optical Emission Spectrum for 100 mTorr Ar

Ar I - 6965.431 Å
Ar I - 7067.218 Å
Ar I - 7147.042 Å
Ar I - 7272.936 Å
Ar I - 7383.980 Å
Ar I - 7503.869 Å
Ar I - 7514.652 Å
Optical Emission Signal (Arb. Units)
0
2000
4000
6000
8000
10000
12000
7000
7100
7200
7300
7400
7500
Wavelength (Angstroms)

Figure 2-26: DARSS Plasma Optical Emission Spectrum for 100 mTorr Ar

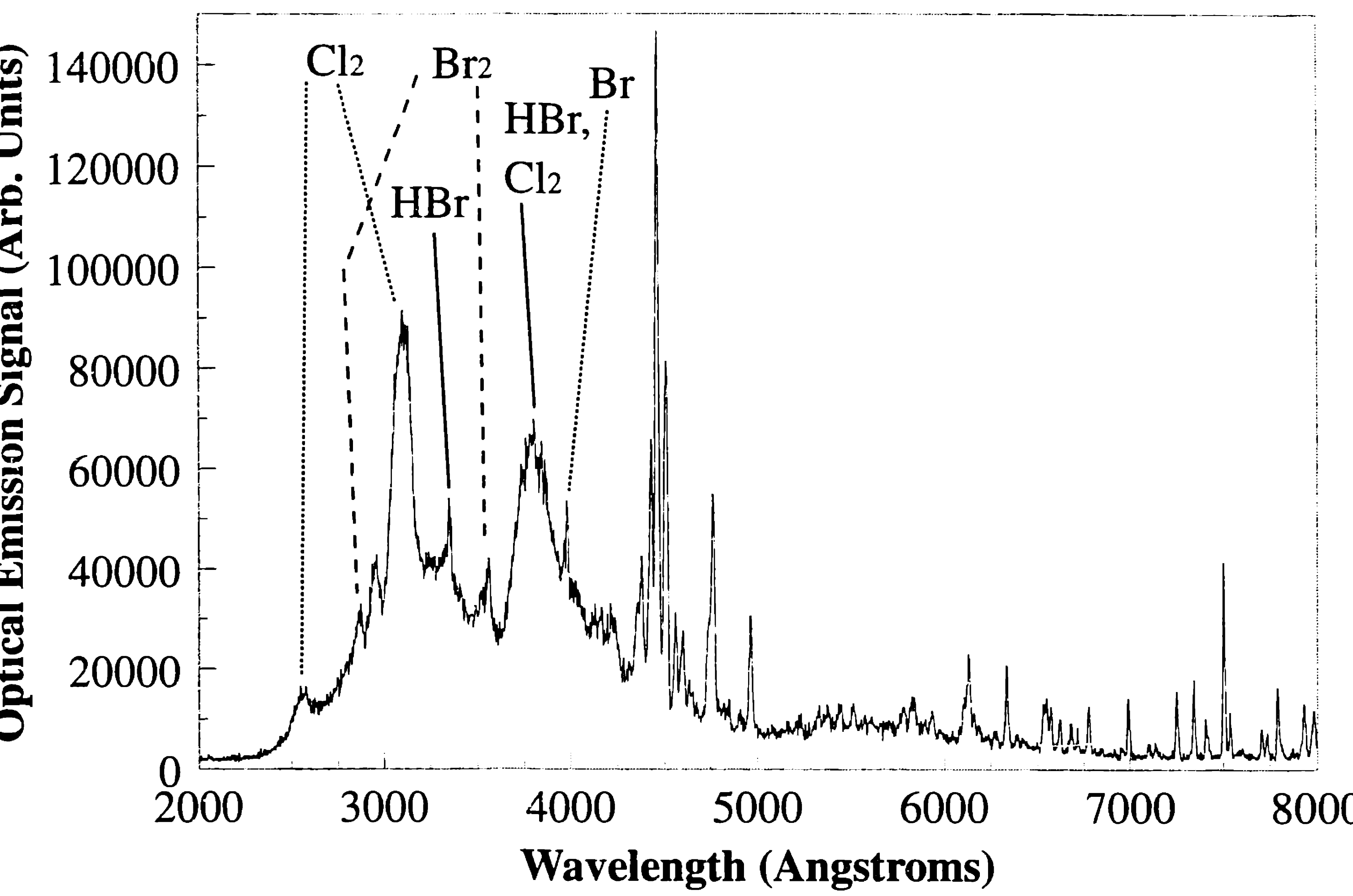


Figure 2-27: Xinix Plasma Optical Emission Spectrum for 100 mTorr HBr (10 SCCM) / Cl_2 (30 SCCM), With Assignments in the UV Region

2-17, and 2-20); this spectrum shows significantly lower resolution than does the spectrum taken with the HR640 (Figures 2-15 and 2-16). Comparison of these figures allows assignment of several features to the component of the feed gas giving rise to them; these assignments for the UV region are shown on Figure 2-27. Assignment in the visible region is easier due to the spectrum being dominated by isolated peaks and not by continua.

2.4.3 Actinometry

The optical emission intensity of a species in the plasma is a complex function of the species density, the electron density, the electron energy distribution function, and the excitation cross-section, as given in Equation 2-2. Thus, changes in the OE intensity may be due to changes in the species concentration, the electron concentration or the electron energy. A simple method to account for changes in the electron density and energy is actinometry [Coburn and Chen, 1980].

A small amount of a noble or unreactive gas [Ar, Xe and N_2 have been used] is added to a reactive plasma. The optical emission from both the reactive species of interest, X, and the noble gas (known as the actinometer) are simultaneously monitored. The emission intensity for the noble gas can be written analogously to Equation 2-2,

$$I_{emis}^{act} = n_e n_{act} A \int \rho_e(E) \, v \, \sigma_{act}(E) \, dE \tag{2-3}$$

Dividing Equation 2-2 by equation 2-3, we get

$$\frac{I^{x}_{emis}}{I^{act}_{emis}} = \frac{n_e n_x A \int \rho_e(E)\, v\, \sigma_x(E)\, dE}{n_e n_{act} A \int \rho_e(E)\, v\, \sigma_{act}(E)\, dE} = \frac{n_x \int \rho_e(E)\, v\, \sigma_x(E)\, dE}{n_{act} \int \rho_e(E)\, v\, \sigma_{act}(E)\, dE} \tag{2-4}$$

If the excitation cross-sections for the actinometer and for the species of interest are similar, the two integrals should be constant, leading to

$$\frac{I^{x}_{emis}}{I^{act}_{emis}} = K_1 \frac{n_x}{n_{act}} \tag{2-5}$$

where K_1 is a constant. For a fixed fraction of actinometer fed into the plasma, n_{act} depends on pressure and temperature. Assuming constant temperature, we can solve for n_x as

$$n_x = \mathrm{P}\, K \frac{I^{x}_{emis}}{I^{act}_{emis}} \tag{2-6}$$

where P is the total pressure and K is a constant.

There are several assumptions implicit in this derivation. First, the dominant method of production for the species of interest must be by electron-impact excitation of a ground state,

$$e^- + X \rightarrow X^* + e^- \tag{2-7}$$

Actinometry has failed to properly measure concentrations for CCl because the dominant production mechanism may be electron-impact dissociation of CCl_x and not electron-impact excitation of CCl [Gottscho and Miller, 1984].

A second assumption is that the dominant quenching mechanism for the excited state is by emission of a photon,

$$X^{*} \rightarrow X + h\nu \tag{2-8}$$

and not by some other mechanism, such as collisional deexcitation.

As mentioned above, the excitation cross-sections are assumed to be similar so that the ratio of the integrals in Equation 2-4 is a constant. If the cross-sections are unknown, lines with similar excitation energy are usually chosen; the 7037.47 Å F emission with an excitation threshold of 14.8 eV and the 7503.869 Å Ar emission with an excitation threshold of 13.5 eV [Kiss, 1991] has been shown to be an accurate predictor of the fluorine density [Coburn and Chen, 1980]. The example of CCl actinometry that failed may be due to the large difference in excitation thresholds; the threshold is ~ 3eV for CCl and ~ 11.5 eV for N_2 [Richards, 1986].

Actinometry has been tried on Cl and Br using Ar previously. Richards [1986], [Richards *et al.*, 1987] compared Cl actinometry using the 8375.94 Å Cl line and the 8115.311 Å Ar line to IR absorption measurement of Cl concentration in a Cl_2 plasma. They determined that actinometry did not work in this case. However, others have shown that actinometry will work if the emission sampled is in the plasma bulk and not in the plasma sheath [Gottscho and Miller, 1984]. Ibbotson *et al.* [1983] compared Br actinometry using the 7005.19 Å Br line and the 7503.869 Å Ar line to IR absorption in a

Br_2 plasma. They concluded that this combination worked well to predict Br concentrations.

Actinometry was attempted on the AME-5000 using Ar and the above mentioned Br and Cl emission lines, along with some others. The Xinix monochromator was unable to accurately resolve Ar emission lines resulting from the addition of 5% (by volumetric flowrate) Ar. Figure 2-28 shows spectra from 7000 Å to 8000 Å with and without Ar. The Ar emissions at 7503.869 Å and 7514.652 Å are lost within the Br emission at 7512.96 Å, and appear as a broadening of the base of the peak. The Ar 7635.106 Å line was distinguishable, though.

A DARSS spectrum from 8000 Å to 8600 Å is shown in Figure 2-29, along with peak identifications for a 100 mTorr HBr/Cl_2 plasma with 5% Ar added. The peaks labeled in bold were used for actinometry. The Ar emission at 8115.311 Å ($4p[5/2] \rightarrow 4s[3/2]^2_0$, emission threshold 13.08 eV) is the only identifiable Ar line and is significantly smaller than the Cl 8375.94 Å ($4p4D^0_{7/2} \rightarrow 4s4P_{5/2}$, emission threshold 10.40 eV) line and the Br 8446.55 Å ($5p4D^0_{3/2} \rightarrow 5s4P_{3/2}$, emission threshold = 9.5149 eV) line. All three of the lines chosen can be unambiguously identified.

The effect of different Ar flowrates and DARSS integration times is shown in Figure 2-30. Although the pressure was constant, the flowrate of Ar (and thus n_{act}) changed. In light of this, the actinometric ratio for Br was calculated as

$$n_{Br} = Q_{Ar}\, K_{Br} \frac{I^{Br}_{8375.94Å}}{I^{Ar}_{8115.311Å}} \tag{2-9}$$

where Q_{Ar} is the flowrate of Argon, and K_{Br} is an unknown constant. The Ar flowrate had

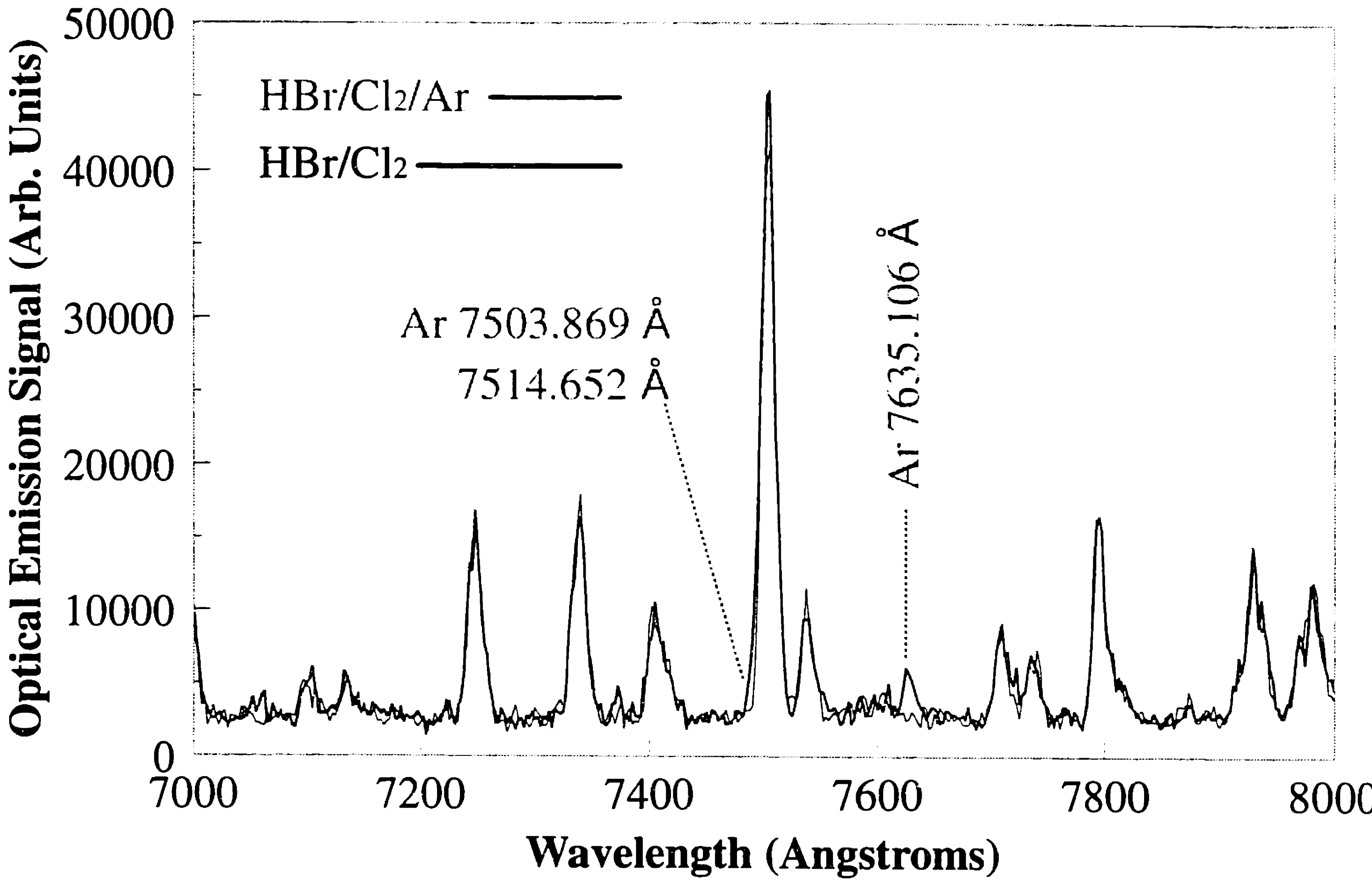


Figure 2-28: Xinix Plasma Optical Emission Spectrum for 100 mTorr HBr (10 SCCM) / Cl_2 (30 SCCM) Without and With Ar (3 SCCM)

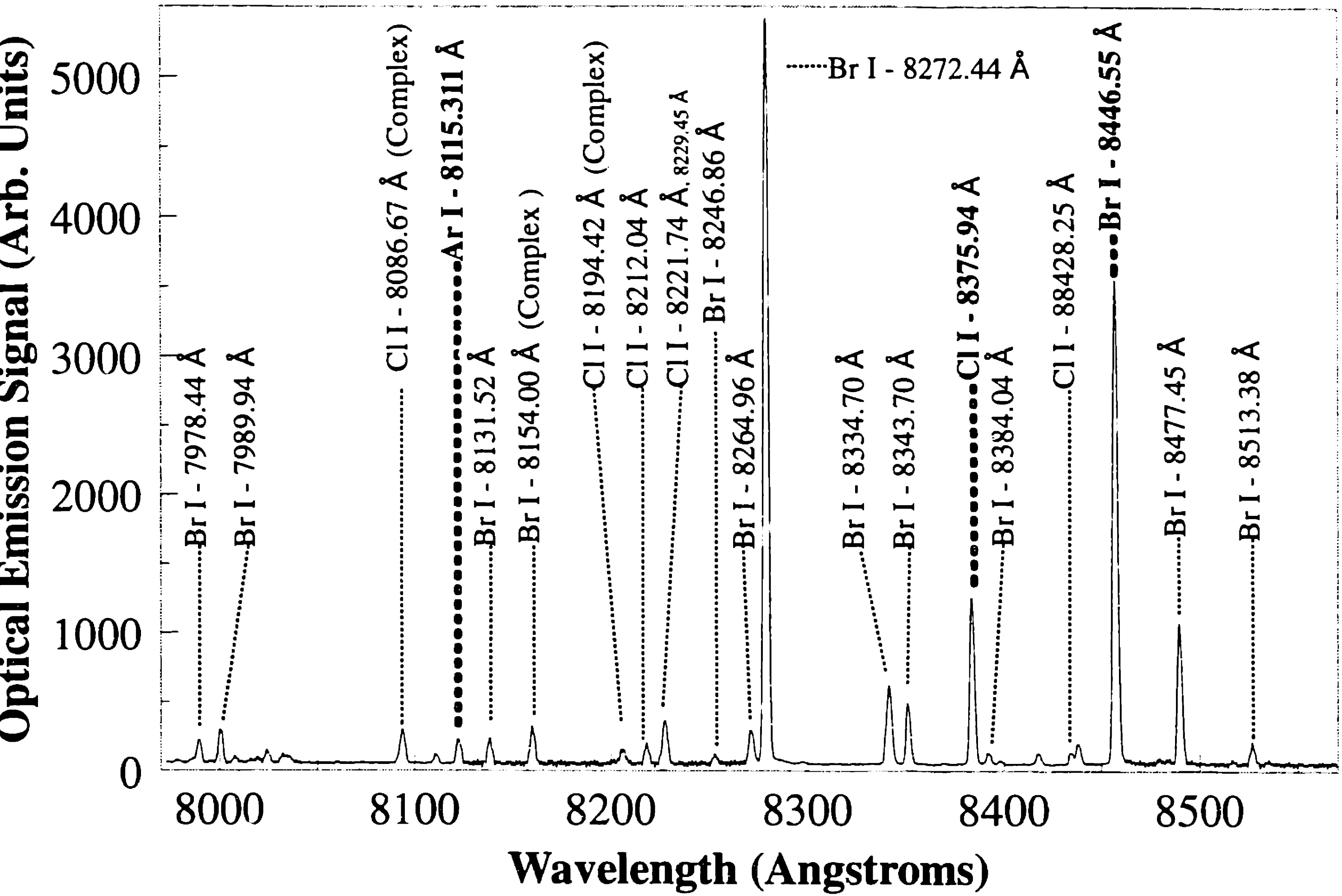


Figure 2-29: DARSS Plasma Optical Emission Spectrum for 100 mTorr HBr (10 SCCM) / Cl2 (30 SCCM) / Ar (2 SCCM) Showing Lines Used in Actinometry

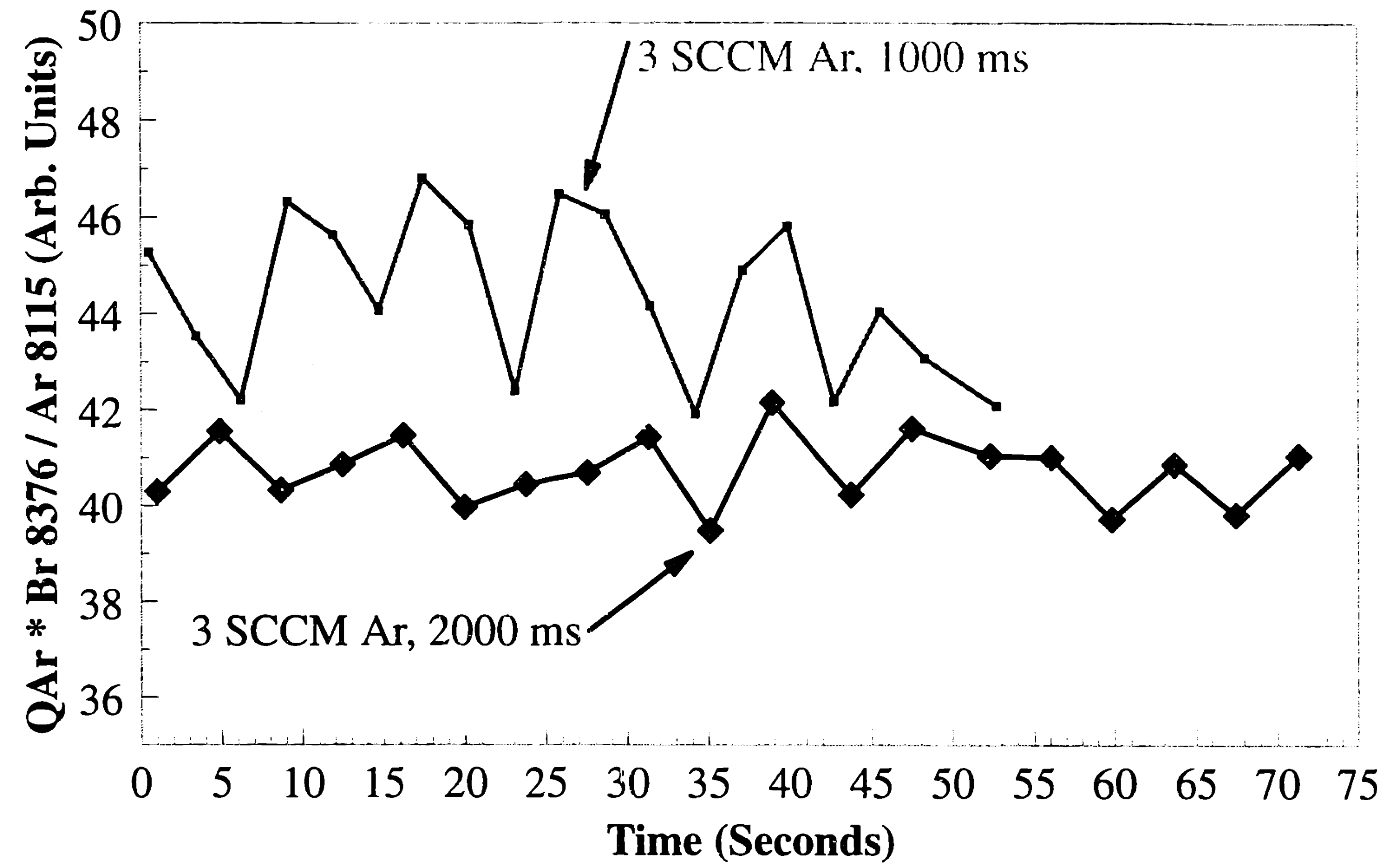


Figure 2-30: Actinometric Ratio (Qar * Br 8375.94 Å/ Ar 8115.311 Å) as a Function of Time for Different Ar Flow Rates (2 SCCM = 5%, 3 SCCM = 7.5%) and DARSS Integration Times.

little effect on the calculated actinometric ratio. For the signals shown in Figure 2-30, at 2 SCCM Ar (5.0%) and 1000 msec integration (light gray) the standard deviation of the signal was 1.8, while for 3 SCCM Ar (7.5%) and 1000 msec integration (dark gray) the standard deviation was 1.7. Operating at either of these Ar flowrates yielded similar results.

The DARSS integration time had a much larger effect. The standard deviation for 3 SCCM Ar and 2000 msec integration (black) was 0.7. The large improvement is due two causes: (1) averaging over a larger number of magnet field periods at 2000 msec. With the field rotating at 2 Hz, a 1000 msec integration averaged over 2 cycles, while a 2000 msec integration averaged over 4 cycles; and (2) more accurate calculation of peak areas due to increased signal intensity. Similar results were obtained for Cl actinometry as well.

A DARSS spectrum from 6950 Å to 7575 Å is shown in Figure 2-31, along with peak identifications for a 100 mTorr HBr/Cl_2 plasma with 7.5% Ar added. The peaks labeled in bold were used for actinometry. The transitions and threshold energies for these lines (along with the ones from Figure 2-29) are summarized in Table 2-6 [Conner, 1991].

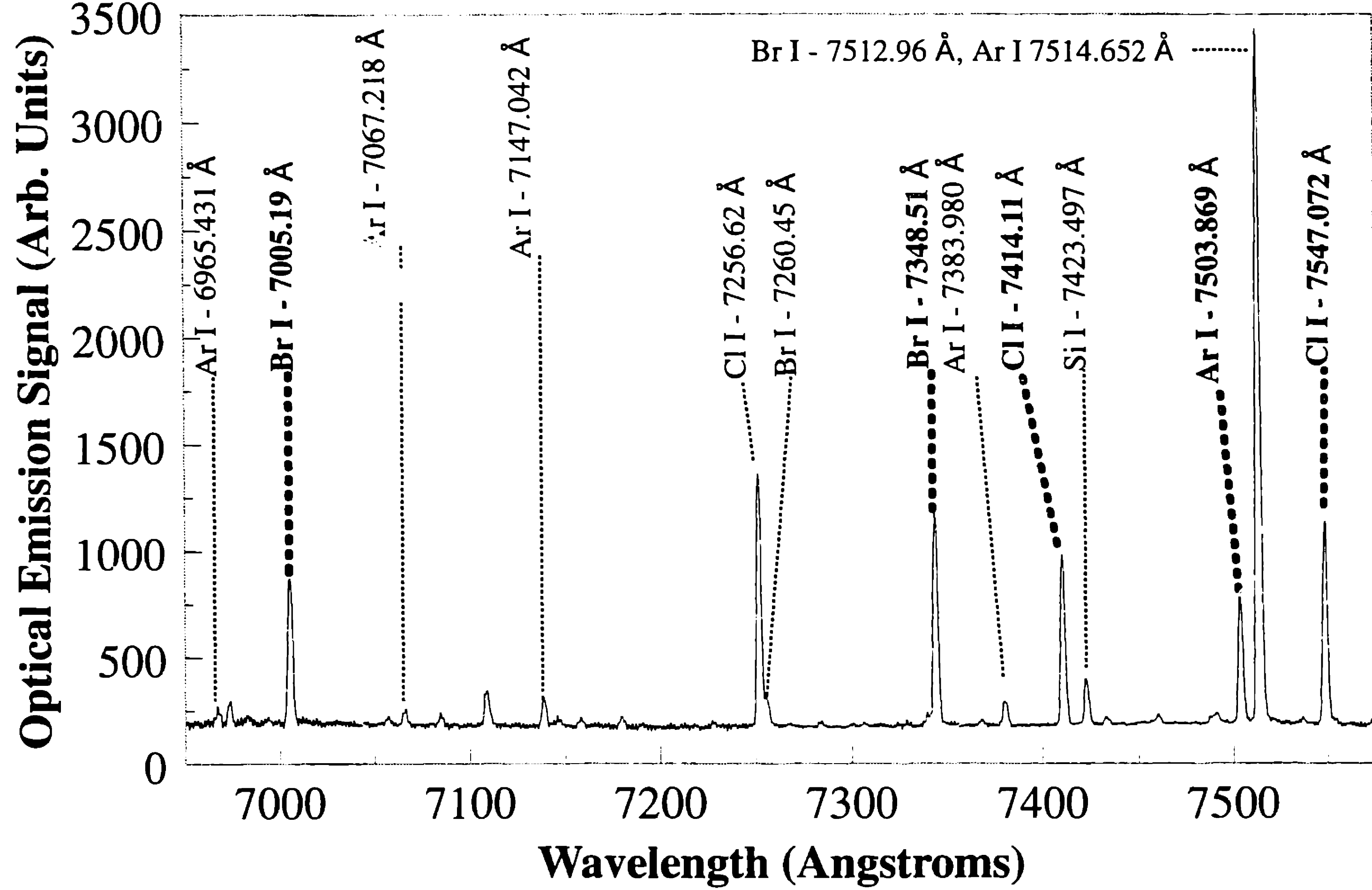


Figure 2-31: DARSS Plasma Optical Emission Spectrum for 100 mTorr HBr (10 SCCM) / Cl2 (30 SCCM) / Ar (2 SCCM) Showing Lines Used in Actinometry

Table 2-6

Electronic Transitions Used in Actinometry

Wavelength (Å)	Lower Level	Energy (eV)	Upper Level	Energy (eV)
Br 8446.55	$5s4P_{3/2}$	8.0474	$5p\ ^4D^0_{3/2}$	9.5149
Br 7348.51	$5s4P_{3/2}$	8.0474	$5p'\ ^2D^0_{5/2}$	9.7342
Br 7005.19	$5s4P_{3/2}$	8.0474	$5p'\ ^4S^0_{3/2}$	9.8168
Cl 8375.94	$4s4P_{5/2}$	8.92	$4p\ ^4D^0_{7/2}$	10.40
Cl 7547.072	$4s4P_{3/2}$	8.98	$4p\ ^4S^0_{3/2}$	10.63
Cl 7414.11	$4s4P_{5/2}$	8.92	$4p\ ^2P^0_{3/2}$	10.59
Ar 8115.311	$4s\ [3/2]^0_2$	11.55	$4p\ [5/2]_3$	13.08
Ar 7503.869	$4s'\ [1/2]^0_1$	11.83	$4p'\ [1/2]_0$	13.48

The two Cl emission lines that were unambiguously identifiable in this region of the spectrum (7414.11 and 7547.072 Å) have similar excitation thresholds to the previously used Cl 8375.94 Å line. Similarly, the Br 7005.19 Å line previously used by Ibbotson *et al.* [1983] is quite similar in threshold to the other Br lines visible here and in Figure 2-29 (7348.51 and 8446.55 Å). Both Ar lines used previously for actinometry are similar, too.

The actinometric ratio (I^{Br}/I^{Ar} and I^{Cl}/I^{Ar}) for two Br and two Cl lines visible in Figure 2-31 were calculated as a function of time (Figure 2-32) using the ratio of peak heights. The two Br lines show similar responses, as do the two Cl lines. However, the data are noisy, due to variations in the peak heights. A much better way of calculating the actinometric ratio is shown in Figure 2-33, where the ratio of peak areas was used. In this case, the trends are much clearer. The significance of this data will be discussed in

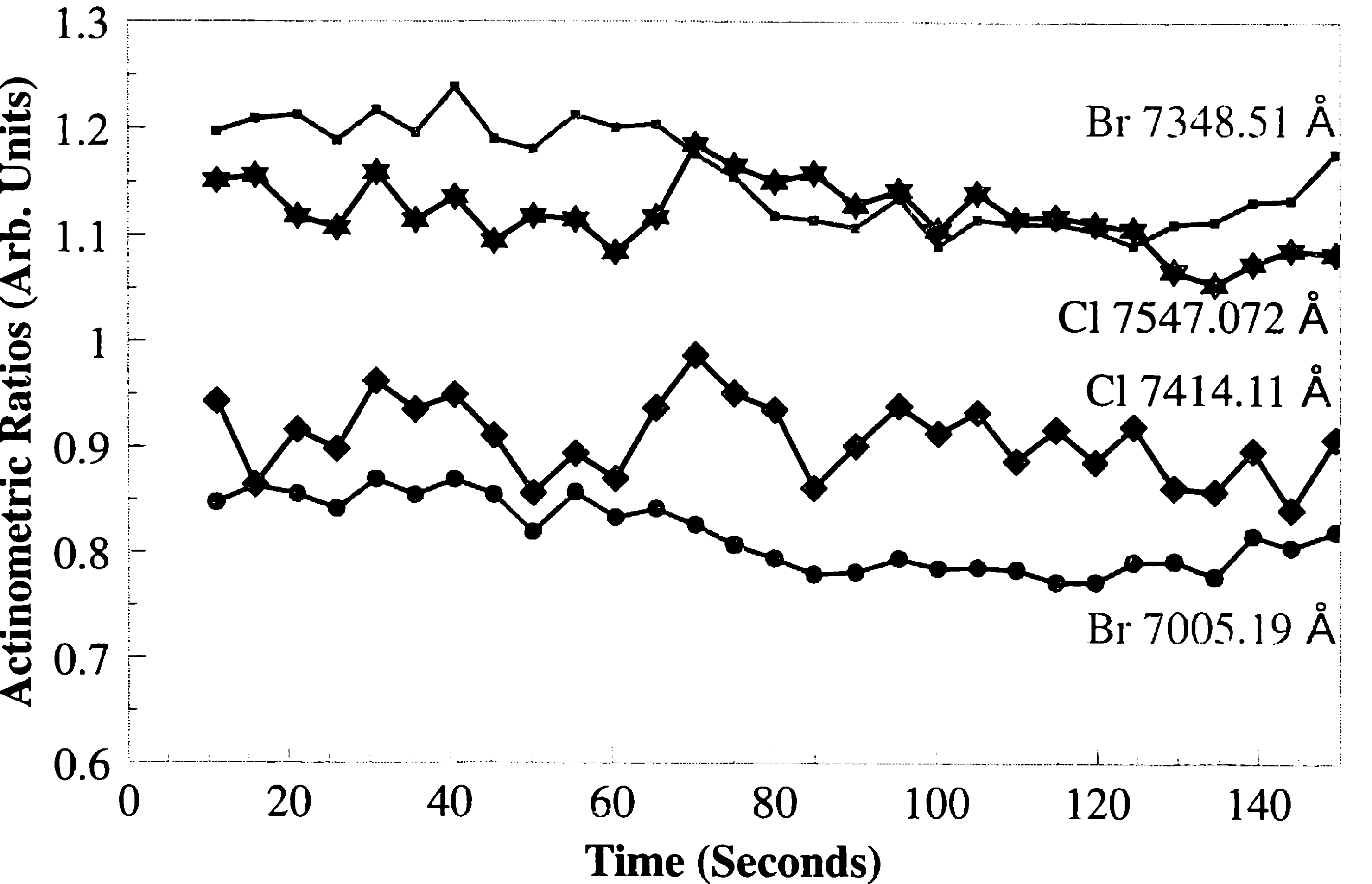


Figure 2-32: Actinometric Ratios Using Peak Heights for 2 Br I and 2 Cl I Lines to Ar I (7503.869 Å) as a Function of Time.

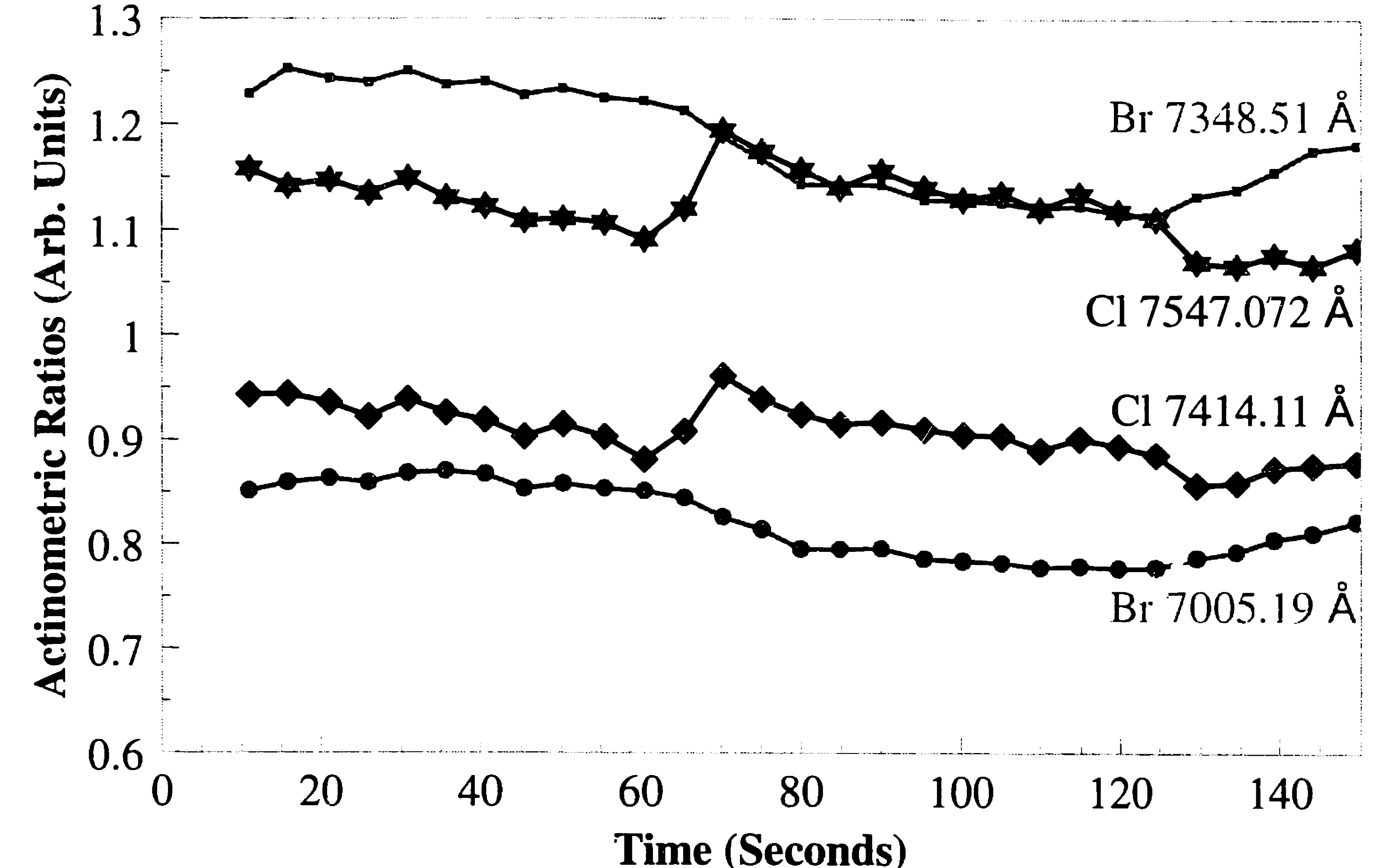


Figure 2-33: Actinometric Ratios Using Peak Areas for 2 Br I and 2 Cl I Lines to Ar I (7503.869 Å) as a Function of Time for the Same Data as Figure 2-32.

Chapter 3. The ratio of peak areas was used in all further actinometric calculations.

2.4.4 End Point

Optical emission is frequently used for end-point determination. For the HBr/Cl_2 chemistry employed in polysilicon etching, Applied Materials recommended using the Br^+ 4704.85 Å emission line ($5s\,^5P \rightarrow 5s\,^5S^0$, [Ibbotson *et al.*, 1983]) for end-point determination [Esselman, 1991]. The OE from this line is plotted as a function of time in Figure 2-34, along with its second derivative (taken with HR640 monochromator). This signal shows three distinct regimes. Initially, there is a large increase in the signal during plasma ignition, followed by a steady state until about 74 seconds, during which time polysilicon is being etched. After this, the signal shows a strong increase as silicon dioxide is being etched (74 - 150 seconds). Finally, the silicon dioxide is etched through, and the signal shows a decrease again as the Si wafer is being etched. As shown here, this line is a good candidate for use as an end-point monitor.

Plasma OE scans from 4600 to 4800 Å are shown in Figure 2-35 for the three monochromators. The Br^+ 4704.85 Å emission line is easily resolved with the HR640 (heavy black line). The Jarrell-Ash with the DARSS can just barely resolve this line, and the Xinix can not resolve it at all. Despite this fact, the Xinix functions very well as an end-point monitor. This is due to an increase in the entire background signal at end-point, which the Xinix detects.

2.5 Etching Test Mask

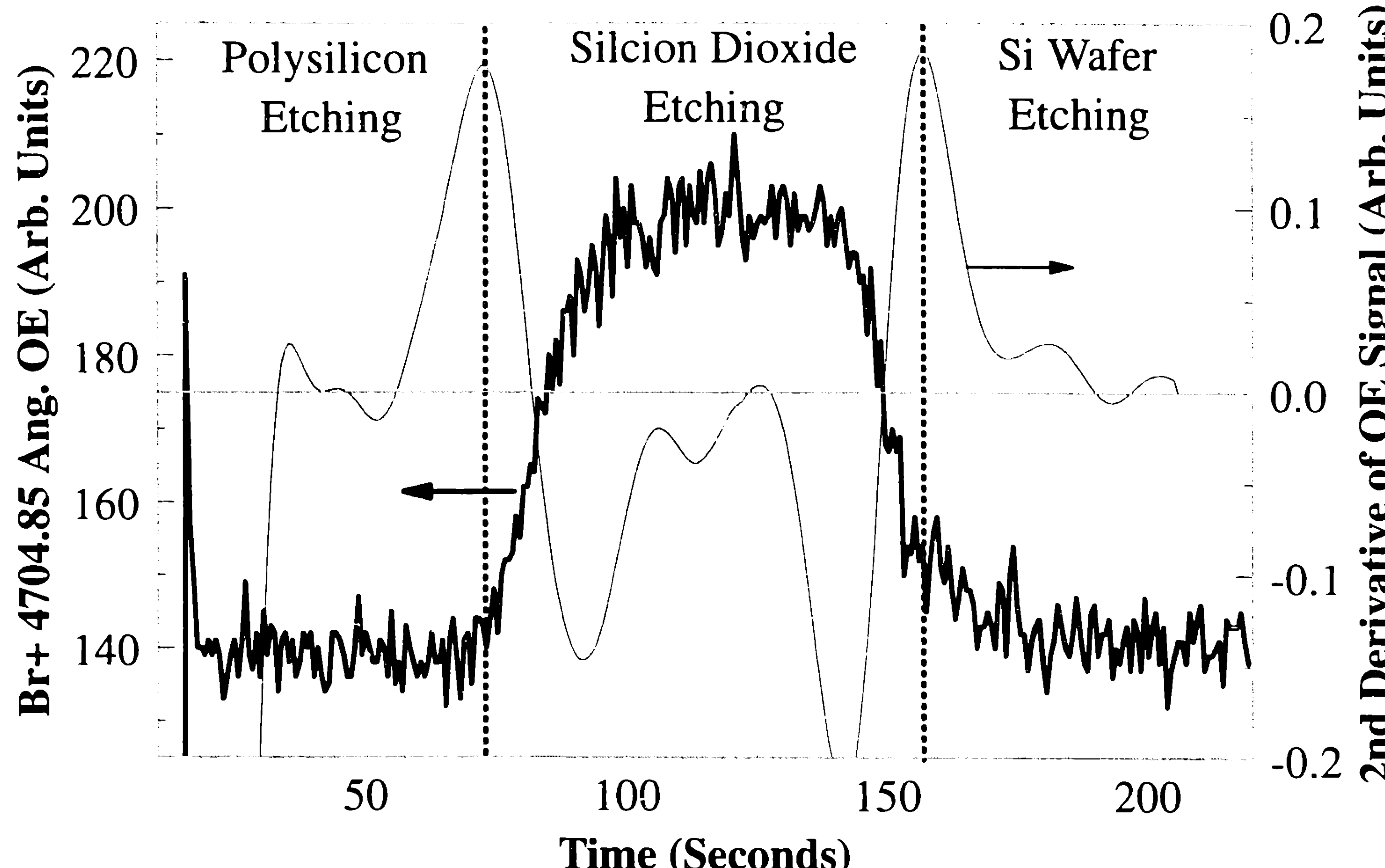


Figure 2-34 : Optical Emission Signal for Br^+ 4704.85 Å Line Used for End-Point Determination. Three Etching Regimes Are Visible: Polysilicon Etching from 15 to 74 Sec, SiO_2 Etching from 74 to 155 Sec, and Si Wafer Etching After 155 Sec.

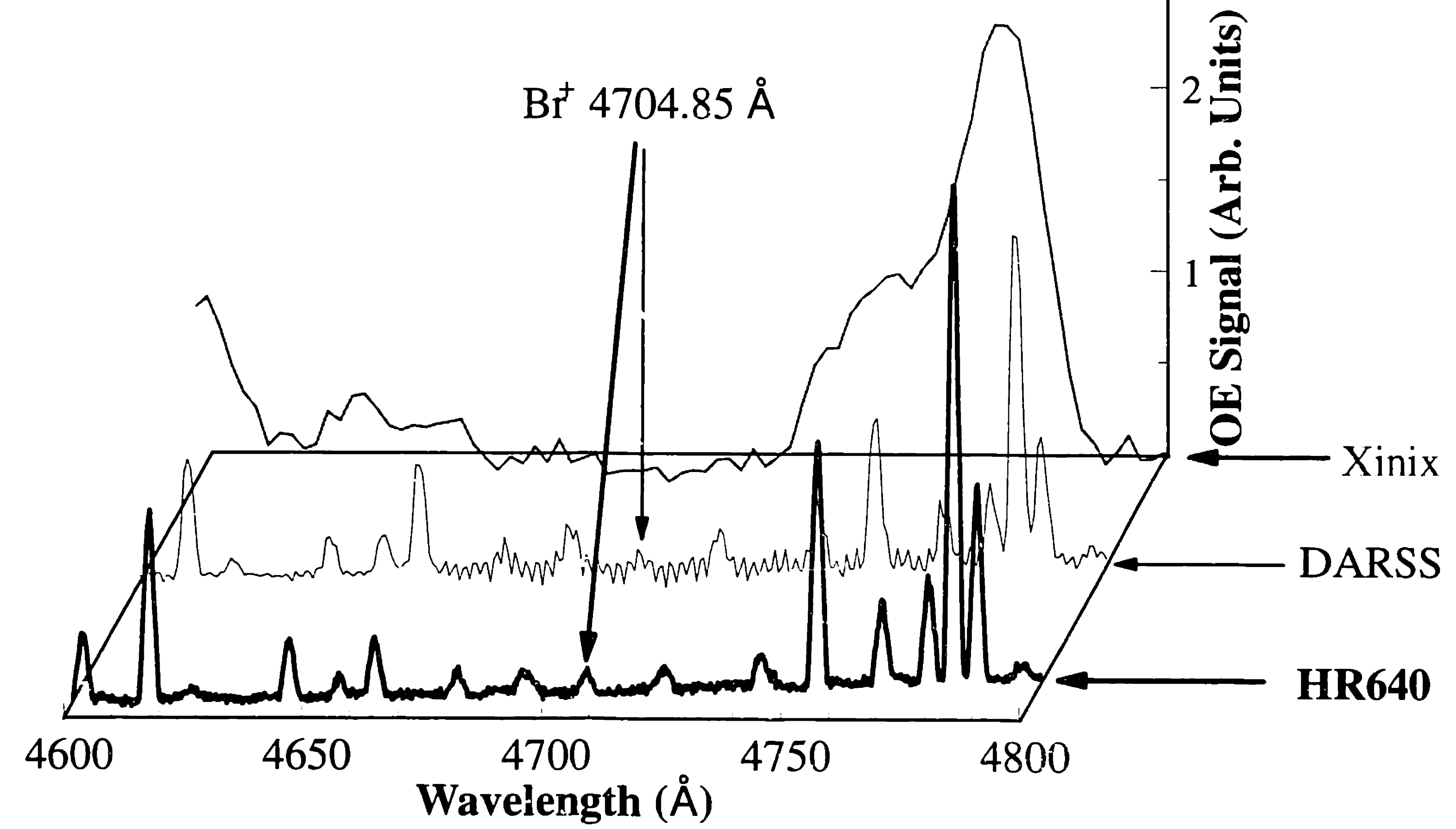


Figure 2-35 : Optical Emission Signal for Br^+ 4704.85 Å Line Used for End-Point Determination On All Three of the Monochromators. The line is Clearly visible on the HR640, Barely Visible on the DARSS and Can Not Be Seen With the Xinix.

A test mask was designed for use in looking for pattern dependencies during plasma etching. A map of this test mask is shown in Figure 2-36. The mask was for use on the GCA stepper located in the ICL (Integrated Circuits Laboratory) in Building 39, and was labeled "SAWIN Mask." When printed, it yielded a 1.0 cm x 1.0 cm die. The mask contains 5 columns by 4 rows of a series of "subdie" regions, replicated across the die. These "subdie" are labeled on Figure 2-36 and are: open (unpatterned regions), line/spacing arrays, serpentine/meandering lines and interdigitated fingers, diffraction gratings, and square vias. Each "subdie" consists of a basic pattern that fills the first quadrant of the "subdie", and is then replicated into the other quadrants by rotating it around the center of the "subdie", with the central corner of the basic pattern pinned to the center of the "subdie." This was done to simplify cleaving the die. Cleaving in any direction will yield cross-sectional views of the features on the die. Similarly, all of the "subdie" patterns are repeated in each row to simplify the cleaving process. Theoretically, the mask contains features as small as 0.5 μm. However, the GCA stepper in the ICL was not capable of printing these lines. Thus, the minimum useable feature size on the test mask was 1.0 μm.

The "Lines" subdie shown on Figure 2-36 consists of a series of arrays of 1000 μm long lines of various line widths, at a constant line spacing. The entire array is repeated with different spacings. The pattern consists of line widths of 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 4.0, 8.0, and 16.0 μm, and line spacings of 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 4.0, 8.0, and 16.0 μm.

The "Vias" subdie consists of rows of square vias (holes). Each row is located 1

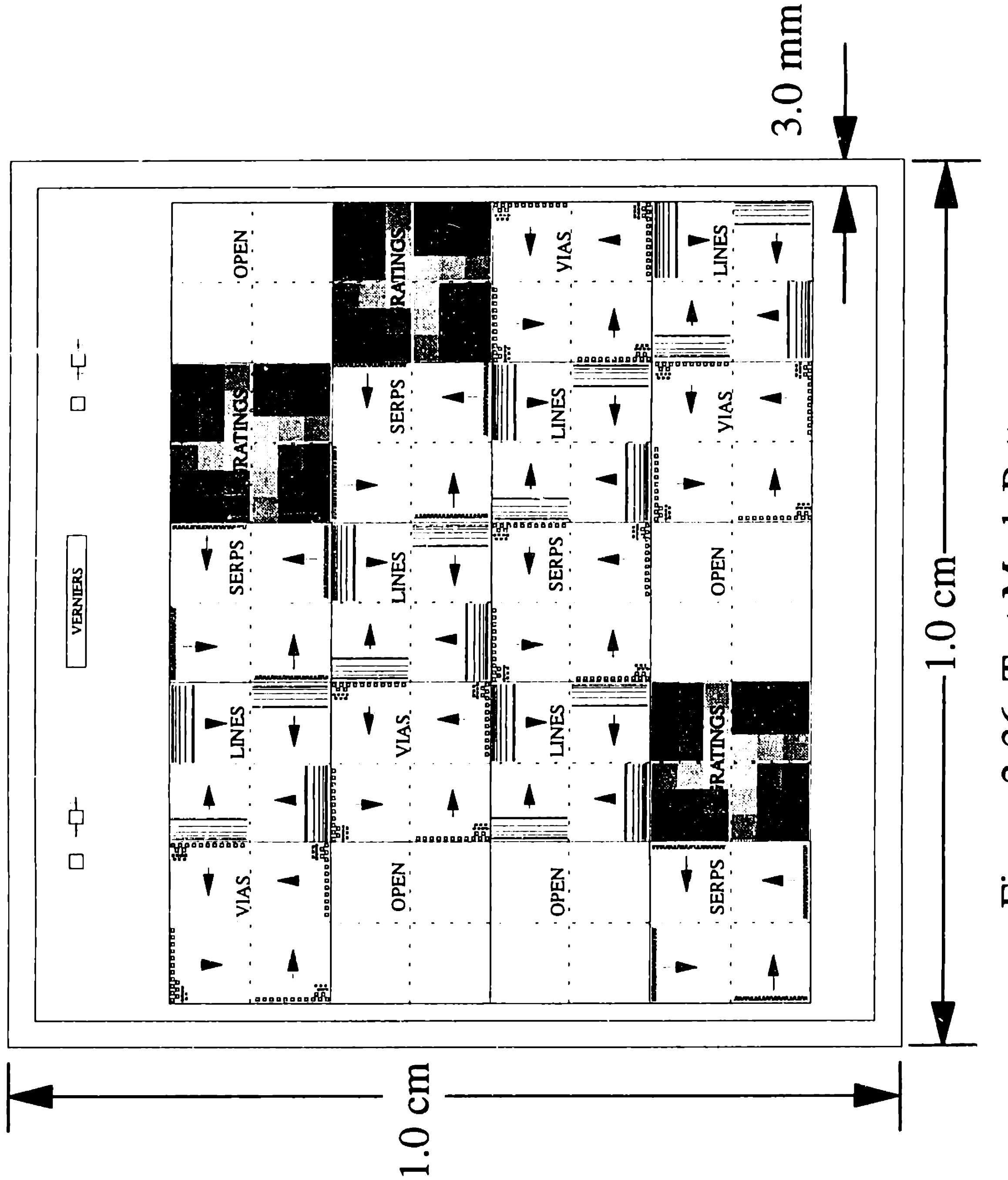


Figure 2-36: Test Mask Pattern.

via size above and below the next row, and is offset by one-half the via size laterally to improve the chance of cleaving through the vias. This subdie contains vias of 1.0, 1.5, 2.0, 3.0, 4.0, 8.0, and 16.0 μm.

The "Serps" subdie consists of a series of serpentine patterns; these are also known as meandering lines or interdigitated fingers. They consist of wandering lines, with the spacing in the wanders equal to the line widths. The line widths are 1.0, 1.5, 2.0, 2.5, 3.0, and 4.0 μm.

The "Gratings" subdie consists of a series of diffraction gratings, *i.e.*, large areas of lines of constant line width and line spacing. A series of 9 gratings makes up the basic pattern that is replicated 4 times on this "subdie." It consists of line widths of 0.5, 1.0, and 2.0 μm and line spacings of 0.5, 1.0, and 2.0 μm.

Finally, the "Open" subdie is a region of unpatterned polysilicon. This pattern was initially included for easy etching rate determination by laser interferometry because the interferometry signal would not be convoluted with that from etching photoresist in this region.

Chapter 3

Characterization of Polysilicon Etching in HBr/Cl_2

3.1 Process Response Analysis

When designing an etching process, one problem is that the number of input variables is quite large. On the AME-5000, a partial list of controlled variables includes: power, pressure, magnetic field strength, magnetic field period, total feed gas flow rate, feed gas composition, cathode temperature, backside He cooling pressure, and wall temperature. On other reactors, the variables would also include: electrode spacing, electrode material, and RF frequency; on the AME-5000, these parameters are fixed. In addition to these controlled variables, there are a number of uncontrolled variables: chamber state (clean/dirty), drift of sensors, and room humidity.

Similarly, when examining the quality of an etch, a large number of possible output metrics exist: selectivity to the underlying layer, selectivity to the mask material, etching profile, etching bias, etching rate, etching rate uniformity, and electrical device quality. The response of these outputs to the controlled variables can be quite complex; nonlinear responses with interactions of the inputs are not uncommon.

3.1.1 Design of Experiments (DOE)

Traditional experimental methods vary one variable at a time, while holding all of

the other inputs constant; this is known as the "one-variable-at-a-time" approach. This method works if the inputs only have a direct effect on the outputs and do not have any interactions. However, if interactions exist, then the measured response of the system to the input variable is a function of the value of the fixed input variables; such interactions can obscure variable effects in this type of experiment.

Statistically designed experiments represent an efficient method for measuring process responses and fitting them to models. The experimental conditions are carefully chosen over a selected parameter space that is to be explored. The models generated using statistically designed experiments allow for accurate determination of process responses using a smaller number of experiments than does traditional experimental methods. These models are useful for process optimization [Sawin and Reif, 1989].

3.1.2 Response Surface Methodology (RSM)

The most frequently-used experimental design methodology is response surface methodology (RSM). This technique has been very well documented [Box *et al.*, 1978] and has been successfully used to plasma etching [Jenkins *et al.*, 1986],[Thompson and Sawin, 1986]. The one significant limitation to RSM is that a functional form for the relationship between inputs and outputs must be assumed [Mocella *et al.*, 1991]. For example, a commonly assumed quadratic model relating two inputs to one output [Box *et al.*, 1978] is given by

$$y = \beta_0 = \beta_1 x_1 + \beta_2 x_2 + \beta_{11} x_1^2 + \beta_{22} x_2^2 + \beta_{12} x_1 x_2 + \epsilon \qquad (3\text{-}1)$$

where y is the output, x_1 and x_2 are the inputs, ϵ is "random, independent experimental error with zero mean and constant variance," and the βs are constants that are determined for the model fit. The model is fit to the data using least squares regression. The form of the model used determines what can be determined from the experimental design. Using a linear model would preclude curvature from the model, while neglecting interaction terms (β_{12}) would eliminate variable interactions. As the number of variables grows, the number of constants in the model grows rapidly.

3.1.3 Neural Networks (NNAPER)

An experimental design method that is free of the assumed-model limitation is a neural network. A neural network is a mathematical model for representing relationships between input and output data in a manner similar to that thought to be used by the human brain. A neural network consists of at least three "layers": input nodes, hidden or connecting nodes, and output nodes (Figure 3-1). Every node in each layer is connected to every node in the next layer by a series of connection weights. The weights adjust the important of the neuron relative to other neurons. The neurons sum up the weights entering them and pass the sum through a "squashing" function, *e.g.*, $y(x) = 1 / (1 + e^{-\alpha x})$.

Neural networks learn by "training". The connection weights are calculated for sets of training data. To test how well a network is trained, "test" data is used to compare the model predictions to experimental measurements. The neural network we employed was written by DuPont Electronics; it is known by its acronym, NNAPER, which stands for: Neural Network Analyzer of Process Evaluation Responses [NNAPER, 1993]. This

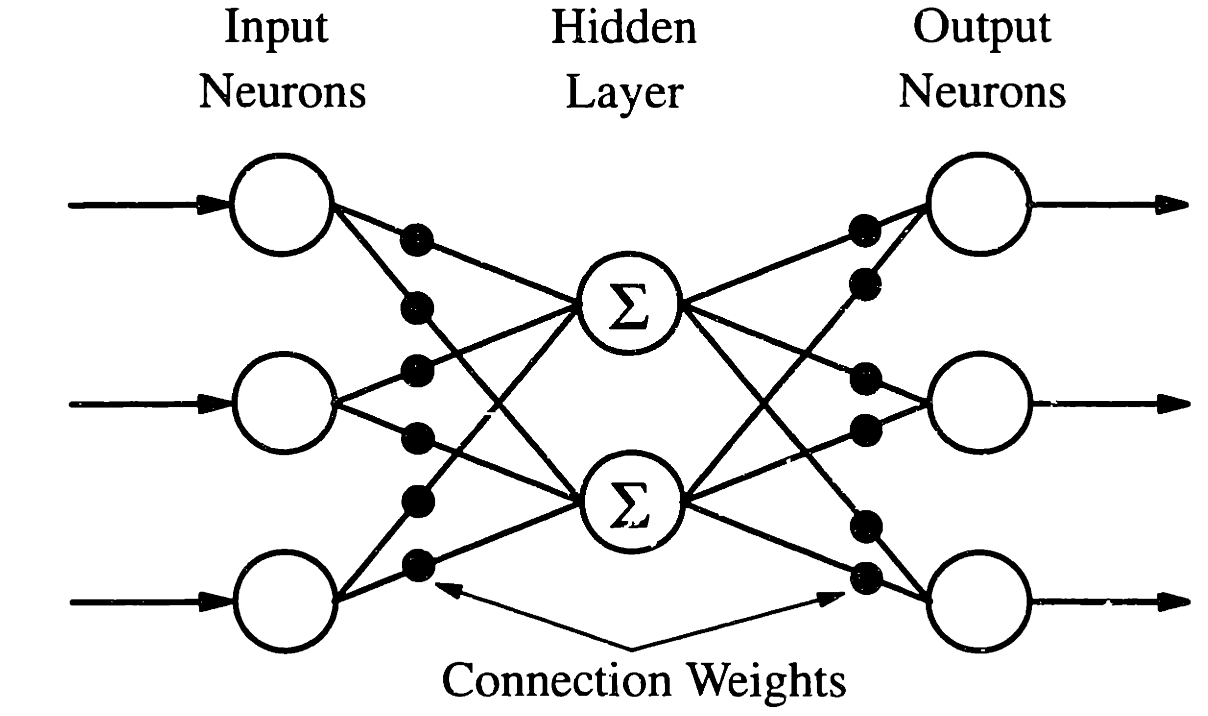


Figure 3-1: Structure of a Three Layer Neural Network. Network consists of Input, Output and Hidden Nodes. After [NNAPER, 1993].

network is a back propagation network. Errors between model and training data are calculated at the output, and propagated back through the network to determine the connection weights in the network. More information on neural networks has been published elsewhere [Owens and Mocella, 1991],[Dalton and Deshmane, 1991],[Bhagat, 1990].

Neural Networks have been used for a number of purposes in the field of plasma etching. Mocella *et al.* [1991] compared an experimental design using NNAPER to one using RSM with RS/1 [BBN, (Cambridge MA)] for polysilicon etching on an AME-5000. They showed that neural networks can be successfully applied to process modeling, with superior results to RSM. Rietman *et al.* [1993] used a neural network to analyze optical emission traces for a polysilicon gate etch to control the remaining oxide thickness using the etch time as the control parameter. Krukar *et al.* [1993] used a neural network to relate etch profile and etch depth to laser light scatterometry data.

3.2 Experimental Design

3.2.1 Experimental Layout

The "design advisor" module of NNAPER was used to construct an experimental design to evaluate polysilicon etching. The controlled parameters were power, feed gas composition, pressure, and magnetic field strength. Their ranges are shown in Table 3-1.

Table 3-1

Designed Experiment Input Parameters and Ranges

Variable	Units	Minimum	Maximum
Power	Watts	125	375
Composition	% HBr	0	50
Pressure	mTorr	20	180
Magnetic Field	Gauss	0	150

The feed gas, a mixture of Cl_2 and HBr, was maintained at 40 SCCM total flowrate. The cathode temperature was 20°C and the wafer backside He pressure was 4 Torr.

The experimental design consisted of a total of 39 experiments. The experimental conditions are listed in Table 3-2. The letters "C," "S," "F," and "T" refer to the type of point the experiment represents, which are: F - Factorial, S - Simplex, C - Center, and T - Test. The significance of where these points lie in the experimental parameter space has been explained elsewhere [Owens and Mocella, 1991]. All of these data points are generated by NNAPER.

Unlike RSM designs, neural network experiments are run in a randomized order. They are run in the order of the chart. This is done for three reasons. First, 3 replicates of the center point are run. This allows a chance to examine experimental error before proceeding too far into the experiments. Second, the full range of parameter space is evaluated to verify that experiments can be conducted over the full range of input parameters. Finally, repeats of the center point over the course of the experiment all for determination of drift in the process.

Table 3-2

Experimental Conditions for Designed Experiments

Number	Type	Power	Compos.	Pressure	Field
1	C	250	25	100	75
2	C	250	25	100	75
3	C	250	25	100	75
4	S	250	0	100	75
5	S	250	25	20	75
6	S	375	25	100	75
7	S	250	25	100	0
8	S	250	25	180	75
9	S	250	25	100	150
10	S	125	25	100	75
11	S	250	50	100	75
12	F	188	12.5	60	113
13	C	250	25	100	75
14	F	188	37.5	140	113
15	T	363	2.5	68	83
16	F	188	12.5	140	113
17	F	313	12.5	140	113
18	T	322	17.5	59	101
19	C	250	25	100	75
20	F	313	12.5	60	38
21	F	313	37.5	60	38
22	F	188	37.5	60	113
23	T	229	35	104	31
24	F	313	37.5	140	38
25	C	250	25	100	75
26	F	188	12.5	140	38
27	T	239	12.5	34	5
28	F	188	37.5	140	38
29	F	313	37.5	60	113
30	F	188	12.5	60	38
31	T	290	27.5	31	137
32	C	250	25	100	75
33	F	313	37.5	140	113
34	T	152	40	122	127
35	F	313	12.5	60	113
36	F	313	12.5	140	38
37	T	225	15	42	53
38	C	250	25	100	75
39	F	188	37.5	60	38

3.2.2 Measurements

Etching experiments were conducted using wafers with undoped polysilicon (5000 Å) over silicon dioxide (1000 Å). Wafers were patterned with 1.1 μm of KTI-820 photoresist. The thickness of the polysilicon film on the wafers was measured at 5 locations (left, center, right, top and bottom) before etching using a Nanospec/AFT 200. The initial film thickness over 50 wafers was found to be 5444 Å with a standard deviation of 24.9 Å. The wafers were etched to remove approximately 4000 Å (4 cycles on the interferometry trace). Film thickness was again measured using the same 5 locations on the wafer. This was used to calculate the etching rate at each of these 5 locations, from which the etching rate uniformity was calculated.

Relative concentrations of Cl and Br were determined using actinometry (Section 2.4.3). Peak areas were calculated for Br 7005.19 Å, Cl 7547.869 Å, and Ar 7503.869 Å emissions. The relative concentrations were calculated according to Equation 2-6. Plasma electrical properties were measured using the Comdel RPM-1.

3.3 Neural Network Construction and Solution

A large number of outputs were available to use in constructing a neural network. The outputs we considered were: etching rate at the left location, etching rate at the center location, etching rate at the right location, etching rate at the top location, etching rate at the bottom location, average etching rate, etching rate uniformity, Cl concentration, Br concentration, real plasma power, RF voltage, RF current, phase angle between current

and voltage, and plasma impedance. A number of different neural networks were constructed to evaluate which of these outputs to use for a final neural network.

The metric used to judge the fit of the network to an individual output was the correlation coefficient, R^2, given by [Mocella *et al.*, 1991]

$$R^2 = 1 - \frac{\Sigma\,[Y_o - Y_p]^2}{\Sigma\,[Y_o^2] - (\Sigma\,[Y_o])^2 / N} \quad (3\text{-}2)$$

where Σ is the sum over all the test ("T") points, Y_o is the observed output, Y_p is predicted output, and N is the number of test points. An R^2 of 1.0 indicates perfect fit of the model to the observed test data, while an R^2 of 0.0 indicates no correlation between the model and the experimental data.

A neural network consisting of the average etching rate and the etching rate uniformity failed to fit the uniformity. For all cases (number of hidden nodes) tried, R^2 for uniformity was 0.0; the fit to the average etching rate was quite good. For this reason, we decided to not use uniformity. Instead, we used the etching rate at all 5 of the measurement locations. From the model results for this data, the uniformity can be calculated.

Similarly, in a neural network based upon the plasma electrical data (actual power, voltage, current, phase angle and impedance) we were unable to fit the impedance. The fit to the plasma power was not as good as the fits for current, voltage, and phase. Since the power can be calculated from these three parameters using

$$\text{Power} = V * I * \cos(\phi) \quad (3\text{-}3)$$

we decided to remove power from the network as well. Calculated values of power were within +0.83%/-0.66% of the powers measured using the RPM. This excellent agreement indicates that the higher harmonics of the 13.56 MHz generator used were small in comparison to the fundamental component [Allen, 1986d].

The lack of fit of a neural network model to uniformity and impedance (and to a lesser extent power) can be explained by examining the type of data used. All of these parameters can be calculated from other more fundamental outputs, *e.g.*, etching rates, voltage, and current. The neural network model provided a good fit to fundamental outputs and not to computed outputs. This has been seen before by others [Mocella, 1991b]; this general design rule will be useful when setting up other experimental designs.

The final neural network was constructed using the 4 input variables of Table 3-1, and 10 output variables: the five etching rates, the Cl and Br concentrations, the RF voltage and current, and the phase angle between the current and voltage.

The one unknown or variable parameter in a neural network is the actual structure of the network, *i.e.*, the number of hidden nodes that gives the best fit of the model to the data. At a small number of hidden nodes, the model complexity is insufficient to fit all of the curvature in the data. At a large number of hidden nodes, the model over-represents the data by fitting noise. The optimum value is determined by (1) maximizing the R^2 of the test data for all of the outputs, and (2) minimizing the RMS error of the test data. The RMS error is defined as

$$\epsilon_{RMS} = \sqrt{\frac{1}{N} \Sigma \, (Y_o - Y_p)^2} \qquad (3\text{-}4)$$

We varied the number of hidden nodes from 7 to 13 (the maximum number allowable); this yielded networks with structures (input-hidden-output) from 4-7-10 to 4-13-10. For each network structure, the model was solved with the 32 "training" points, and then the predictions were compared to the 7 "test" points. After this, the 39 points were merged into one large data set and the model retrained using the complete data set. At this point, comparisons between the test data and the model predictions were not meaningful because the test data was used to construct the model.

The RMS error for both the training and test data is shown in Figure 3-2. The training error is plotted at 5 times its value to make it comparable to the test error. As the number of hidden nodes increases, the training error decreases This is due to the increased fit of the neural network to the data. This is similar to curve fitting with polynomials; increasing the degree of the polynomial increases how well the data can be fit. The behavior of the training error is in contrast to the test error. As the number of hidden nodes increases, the test error goes through a minimum at 10 nodes, and then increases again. This is due to the model being too simple to capture all of the curvature in the data at low hidden nodes (7) and the model fitting the noise in the data at high hidden nodes (13). This plot indicates that a network with 10 hidden nodes appears to be the correct choice. To verify this, the R^2 values for the 10 outputs have to be examined. The large spike in the figure at 11 hidden nodes is due to a poor model fit. The response of the neural network is nonlinear. The 4-11-10 structure was not capable of fitting this data set. Changing the convergence criterion for the network did not change the calculated results for this structure.

The correlation coefficients for the five polysilicon etching rates (Figure 3-3)

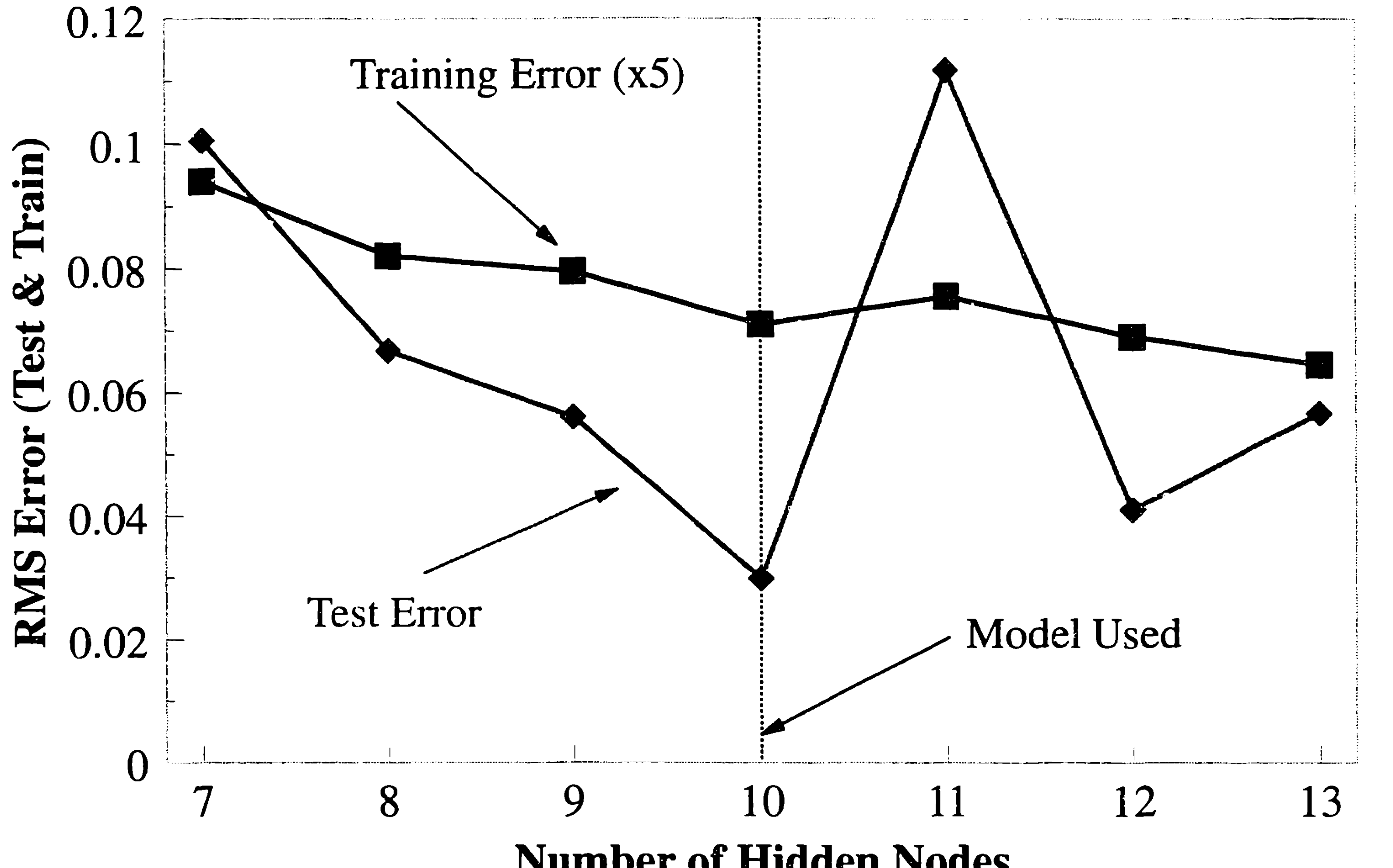


Figure 3-2: Evaluation of Neural Network Model Fit to Training and Test Data. Training Error Shows Model Fit to Training Data; Test Error Shows Model Fit to Independent Test Data Used to Evaluate Fit. Dashed Line Shows Model Used.

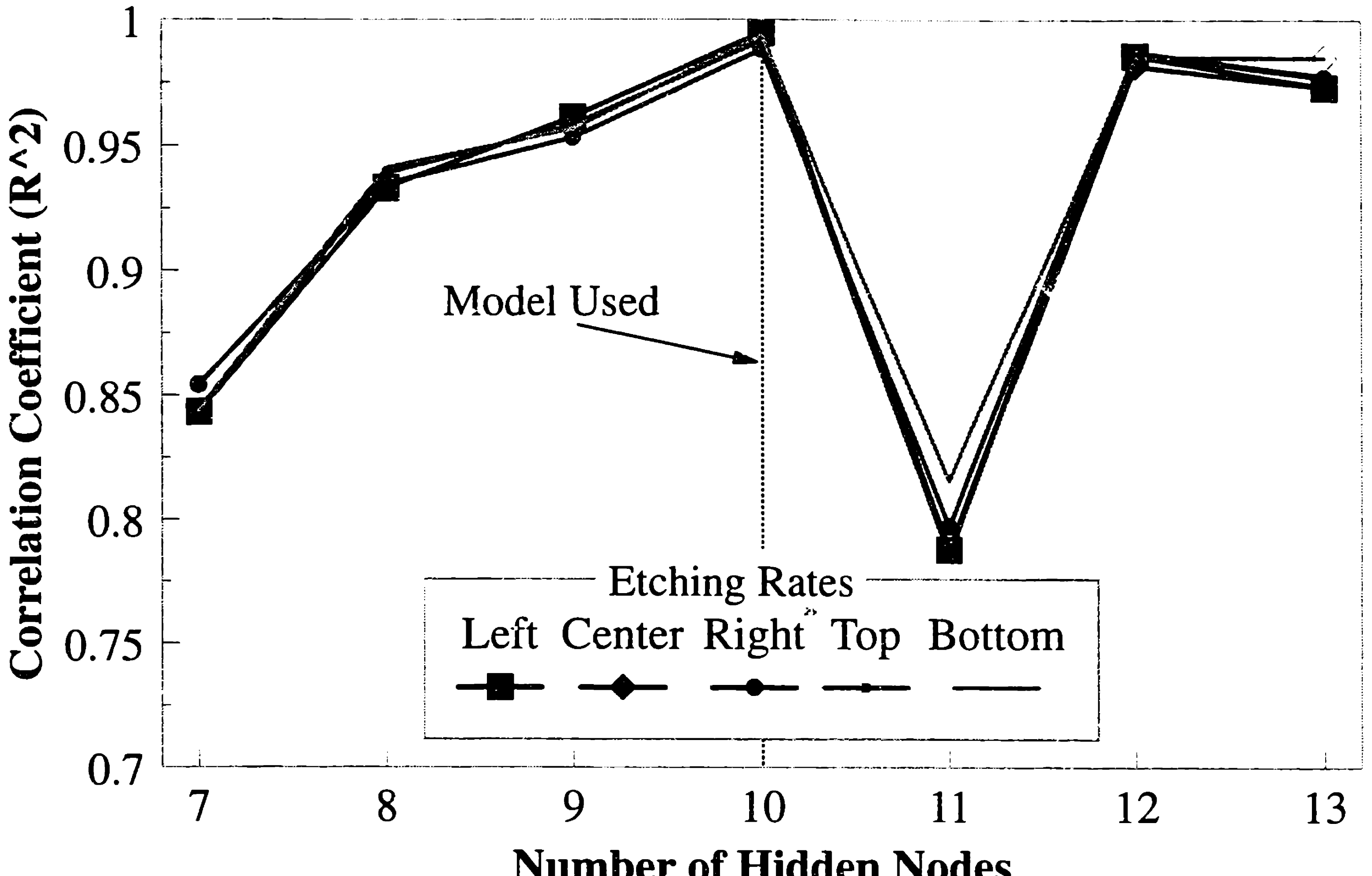


Figure 3-3: Evaluation of Neural Network Model Fit to Polysilicon Positional Etching Rate Data (Uniformity).

showed behavior similar to that of the test error. R^2 increased with an increasing the number of hidden nodes from 7 to 10. It was maximum at 10 nodes, with an average value of 0.993. R^2 decreased as the number of hidden nodes increased beyond 10. The explanation for this behavior is identical to that given in the previous paragraph. The correlation coefficients for the other 5 outputs (the Br and Cl concentrations, voltage, current and phase angle) also followed the above trends (Figure 3-4). At 10 hidden nodes, the values were 0.970, 0.877, 0.980, 0.986, and 0.987, respectively. The Cl concentration consistently showed the lowest fit. All of these results indicated that the optimum structure for the neural network was 4-10-10. Therefore, that structure was used for the remainder of this chapter.

3.4 Plasma Electrical Parameter Characterization

Plasma electrical properties are both easy to measure and insightful. A simple electrical analog model provides estimates of the time-averaged values for a number of plasma properties.

3.4.1 Electrical Analog Model

Allen [1986d] derived a number of plasma properties from electrical measurements using a simple electrical analog model. This model represented the plasma as a series of electrical components (Figure 3-5). The plasma bulk was modeled as a resistor, R_b, to simulate the resistance to the flow of an electron current through the bulk.

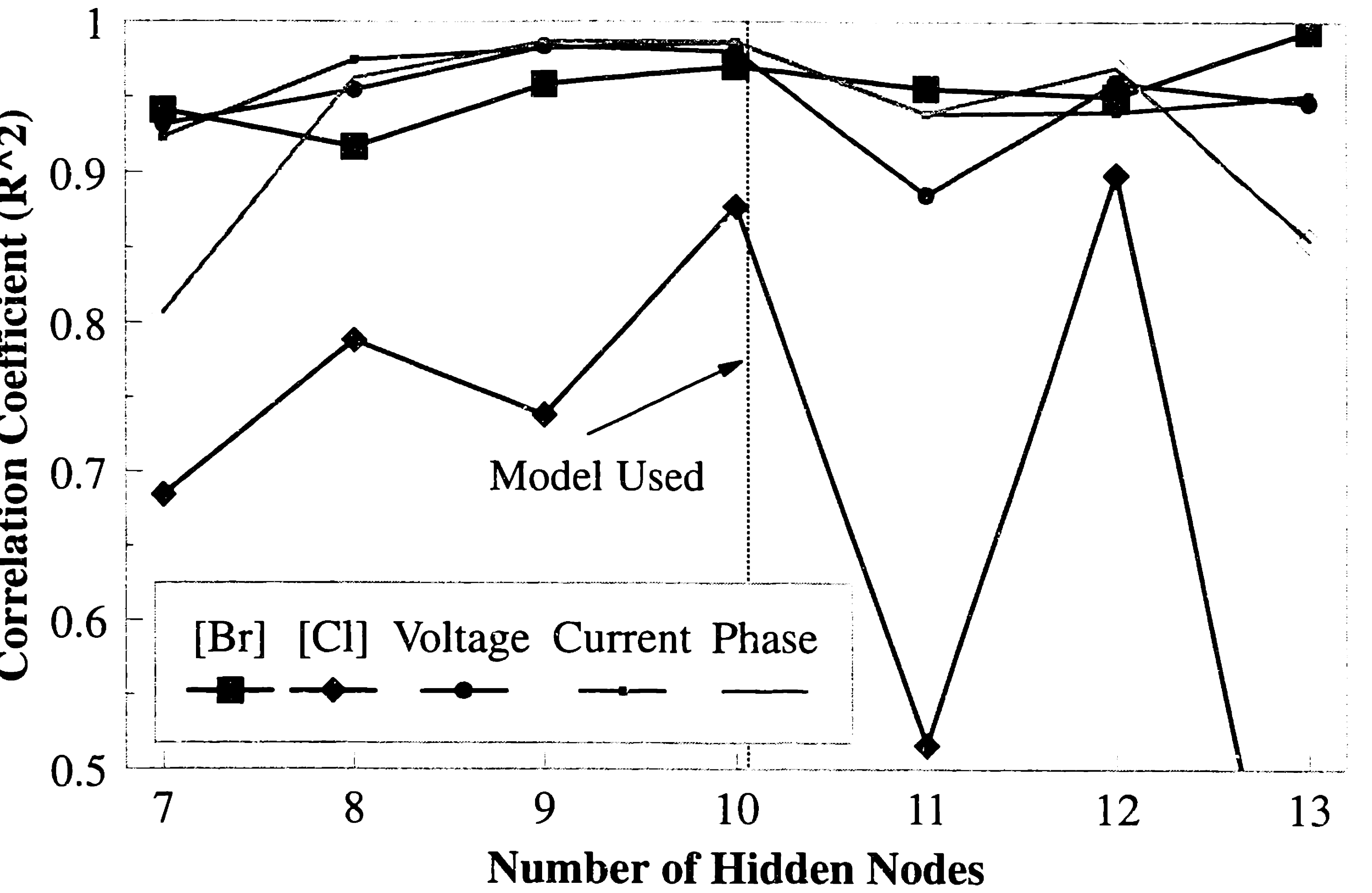


Figure 3-4: Evaluation of Neural Network Model Fit to Polysilicon Etching Data: Etchant Concentrations from Actinometry and Electrical Parameters from RPM.

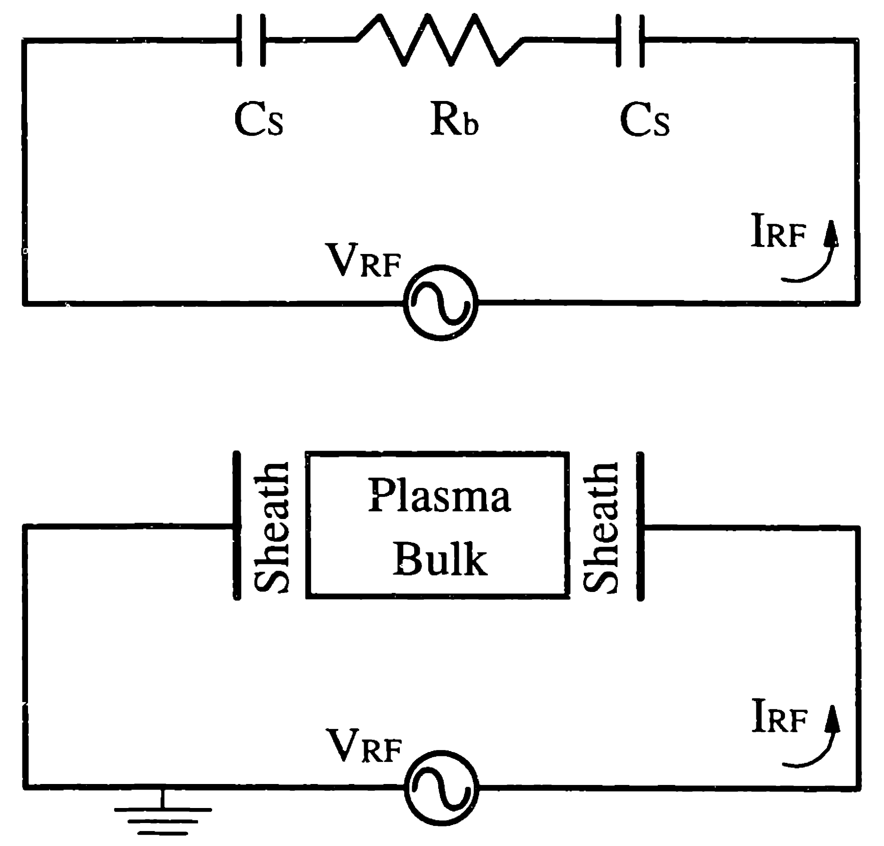


Figure 3-5: Plasma Electrical Analog Model (After [Allen, 1986d]).

The sheath regions, where the high fields repel electrons, was modeled as capacitors, C_s, to simulate the displacement current there.

The model is for a symmetric discharge. In an asymmetric discharge, such as the AME-5000, the sheath sizes are very different. This difference in sheath sizes leads to two different sheath capacitances, C_{s1} and C_{s2}. However, one of the sheaths will be much thinner than the other; the sheath at the smaller electrode - in this case the powered cathode. The majority of the capacitance of the plasma is from the thinner sheath. Therefore, we modified the model of Allen [1986d] to have a resistor and a single capacitor representing the cathode sheath.

The values for R_b and C_s were found using an impedance analysis. They are given by:

$$R_b = \frac{|V|}{|I|} \cos(\phi) \tag{3-5}$$

$$C_s = \frac{|I|}{\omega \, |V| \sin(\phi)} \tag{3-6}$$

where $|V|$ is the magnitude of the RF voltage, $|I|$ is the magnitude of the RF current, ϕ is the phase angle between current and voltage, f is the generator frequency (13.56 MHz), and $\omega = 2\pi f$.

Allen [1986d] showed that the electron density, n_e, could be calculated from the electrical analog model with a few simplifying assumptions: (1) all plasma power is dissipated in the bulk; (2) the electron density, n_e, is uniform in the bulk; (3) the bulk current is carried by electrons; and (4) the electron motion is mobility limited.

$$n_e = \frac{d}{A\, e\, R_b\, \mu_e} \qquad \textbf{(3-7)}$$

where n_e is the time averaged electron density, d is the plasma thickness, A is the plasma area, e is the electronic charge, and μ_e is the electron mobility. The electron mobility can be calculated from the following correlation (to a first approximation) [Richards, 1986]

$$\mu_e P = 1.4 \times 10^5 \ cm^2\text{-}Torr/V\text{-}s \qquad \textbf{(3-8)}$$

where P is the pressure.

Another important plasma parameter that was derived from the electrical model was the electric field to pressure ratio, E_b/P, which is important in characterizing electron energy distributions. The derived equation [Allen, 1986d] is

$$\frac{E_b}{P} = \frac{R_b\, |I|}{\sqrt{2}\, P\, d} \qquad \textbf{(3-9)}$$

where E_b/P is the rms bulk electric field to pressure ratio [V/cm-Torr], and the other parameters are as defined above.

A final plasma parameter that was estimated from the electrical model was the RF sheath voltage. This is important because it gives an estimate of the maximum ion energy that will strike the surface (in symmetric systems). It can be combined with measured DC bias to estimate the maximum ion energy in an asymmetric system. The estimate was

$$V_{s,RF} = \frac{|I|}{\omega\, C_s} \qquad \textbf{(3-10)}$$

3.4.2 Results and Discussion

The plasma electrical properties measured with the Comdel RPM-1 are listed in Table 3-3, along with the plasma properties calculated using the electrical analog model: R_b, C_s, E_b/P, $V_{s,RF}$, and n_e. The experimental run numbers listed in this table correspond to the experimental conditions of Table 3-2.

The first thing to notice in the data of Table 3-3 are the values of ϕ. The phase shift over the 39 experiments varied between -84.0° and -86.0°, with an average of -86.3°. These very large negative shifts indicate that the plasma was very capacitive; at these extreme angles, cos(ϕ) was on the order 1 and sin(ϕ) was approximately ϕ (radians). Measurement errors directly affected the calculated value of C_s. Due to the very large phase shift, the calculated values for the bulk resistance, R_b, were quite small; they ranged between 1.1 and 3.1 Ω. The sheath capacitances were estimated to be on the order of 375 pF and varied only slightly with operating conditions. The RMS values of the sheath voltage, $V_{s,RF}$, were quite close to the RMS driving voltage due to the high phase shift. Electron densities were on the order of 10^{10} cm^{-3}, and will be discussed in more detail shortly. The rms values of the bulk electric field to pressure ratio ranged between 14.4 and 103.6 V cm^{-1} $Torr^{-1}$. The low bulk resistance led to some seemingly low E_b/P values. In an Ar discharge, values that low would have been close to extinction.

The neural network fit to the RF voltage data is shown in Figure 3-6. The RMS voltage is plotted as a function of the generator voltage and the magnetic field strength, for the centerpoint conditions (pressure = 100 mTorr, and composition = 25% HBr (30 SCCM Cl_2 / 10 SCCM HBr)). The RF voltage varies greatly over the range of power and magnetic field, between a maximum of 416 and a minimum of 146 Volts. As expected, the voltage increases with increasing power, and decreases with increasing magnetic

Table 3-3

Plasma Electrical Property Characterization

Run	Power (Watts)	|V|rms (Volts)	|I|rms (Amps)	ϕ (°)	R_b (Ω)	C_s (pF)	n_e (cm^{-3})	E_b/P (V/cm-Torr)	$V_{s,RF}$ (Volts)
1	181.23	275.5	9.13	-85.89	2.17	389.97	1.90e+10	25.94	274.79
2	181.40	275.9	9.13	-85.89	2.17	389.4	1.90e+10	25.94	275.19
3	181.56	276.2	9.13	-85.89	2.17	388.98	1.90e+10	25.97	275.49
4	184.43	259.4	8.63	-85.29	2.47	391.8	1.67e+10	27.93	258.53
5	185.82	327.8	10.62	-86.92	1.66	380.81	4.96e+09	115.57	327.33
6	289.79	364.1	11.96	-86.2	2.02	386.39	2.04e+10	31.67	363.3
7	187.47	340.9	10.87	-87.1	1.59	374.73	2.59e+10	22.63	340.46
8	178.00	244.6	8.15	-84.9	2.67	392.63	2.77e+10	15.87	243.63
9	176.95	229.7	7.78	-84.31	2.93	399.5	1.40e+10	29.87	228.57
10	77.72	166.9	5.5	-85.14	2.57	388.18	1.60e+10	18.54	166.3
11	181.81	286.9	9.47	-86.19	2.01	388.27	2.04e+10	25	286.27
12	127.28	211.6	7.11	-85.15	2.52	395.8	9.80e+09	39.15	210.84
13	181.30	276	9.13	-85.89	2.17	389.26	1.90e+10	25.95	275.29
14	128.93	199.4	6.67	-84.42	2.91	394.48	1.98e+10	18.16	198.46
15	274.75	347	11.49	-86.04	2.08	389.57	1.34e+10	46.21	346.17
16	132.28	187.4	6.21	-83.48	3.43	391.47	1.68e+10	19.95	186.19
17	228.31	259.2	8.78	-84.22	2.97	399.61	1.94e+10	24.47	257.88
18	240.81	322.6	10.72	-86	2.1	390.98	1.16e+10	50.05	321.81
19	180.00	277	9.16	-85.92	2.15	389.12	1.91e+10	25.89	276.3
20	245.52	379.5	12.14	-86.93	1.67	376	1.47e+10	44.43	378.96
21	251.43	402.4	12.68	-87.2	1.55	370.29	1.59e+10	43.03	401.92
22	127.84	222.1	7.44	-85.58	2.3	394.35	1.07e+10	37.47	221.44
23	168.13	308	9.92	-86.83	1.72	378.61	2.49e+10	21.52	307.53
24	237.98	350.3	11.36	-86.6	1.83	381.3	3.15e+10	19.47	349.68
25	179.91	276.6	9.14	-85.91	2.16	388.83	1.90e+10	25.91	275.89
26	130.14	224.3	7.38	-85.5	2.38	387.37	2.41e+10	16.5	223.61
27	195.01	398.2	12.15	-87.7	1.32	358.41	1.06e+10	61.68	397.88
28	130.32	247.2	8.04	-86.22	2.02	332.57	2.84e+10	15.26	246.66
29	231.65	314.3	10.54	-86	2.08	394.56	1.19e+10	47.95	313.53
30	131.11	268.9	8.7	-86.8	1.73	380.34	1.43e+10	32.83	268.48
31	211.21	289.5	9.82	-85.73	2.19	399.24	5.81e+09	91.21	288.7
32	180.59	277.12	9.16	-85.91	2.16	388.95	1.90e+10	25.96	276.41
33	229.39	284.7	9.6	-85.2	2.48	397.16	2.32e+10	22.34	283.7
34	99.75	168.2	5.61	-83.93	3.17	393.67	1.58e+10	19.12	167.26
35	230.97	301.6	10.13	-85.69	2.24	395.34	1.10e+10	49.58	300.75
36	232.34	312.8	10.32	-85.89	2.17	388.23	2.65e+10	21.04	311.99
37	163.74	305.22	9.86	-86.9	1.67	379.72	1.03e+10	51.57	304.77
38	179.34	275.5	9.11	-85.9	2.16	389.11	1.90e+10	25.84	274.8
39	134.32	287.84	9.22	-87.10	1.58	376.44	1.56e+10	31.85	287.47

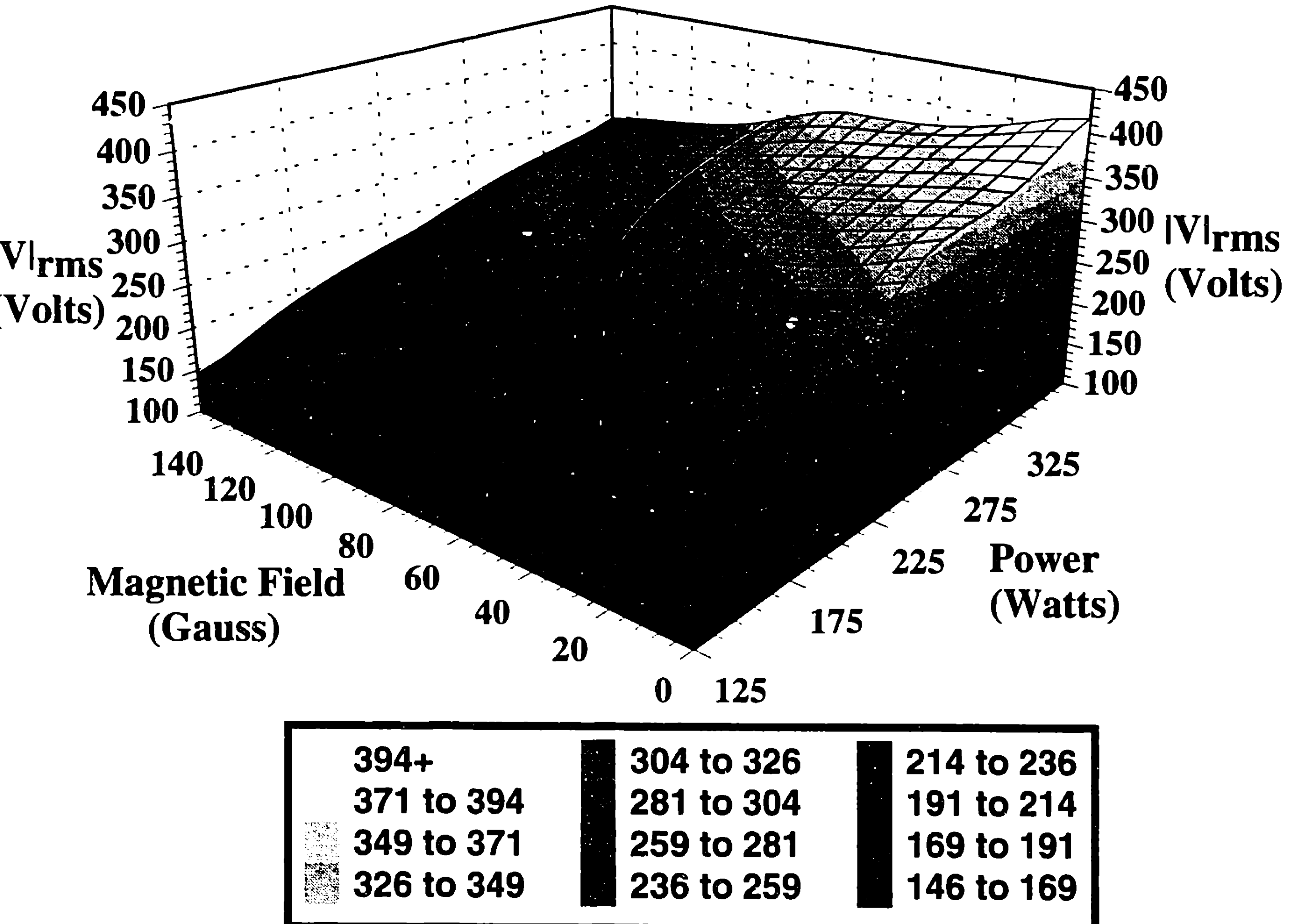


Figure 3-6: Modeled rms RF Voltage as a Function of Power and Magnetic Field Strength. Pressure = 100 mTorr, Composition = 25% HBr.

field. Very similar results were obtained for the current as well (Figure 3-7). The current and voltage both showed similar responses to pressure and feed gas composition; both current and voltage increased with decreasing pressure and with increasing HBr content in the feed gas.

The phase shift, ϕ, between the current and voltage waveforms is less intuitive than are the response of the current and voltage (Figure 3-8). ϕ becomes slightly more negative as power in increased; the magnetic field has a large effect on ϕ, with ϕ becoming less negative as the magnetic field strength increases, indicating that the plasma is becoming less capacitive and more resistive as the magnetic field increases.

The electron density was calculated from Equation 3-7; the components d, A, e, and μ_e are all independent of magnetic field and power. Thus, changes in n_e arose only changes in R_b. The electron density is plotted as a function of power and magnetic field in Figure 3-9 for the same conditions as Figures 3-6, 3-7 and 3-8. The order of magnitude for n_e is correct, 10^{10} cm^{-3}. However, two disturbing features are visible in this plot. The first is that n_e decreases slightly with increasing power. The second is that n_e decreases with increasing magnetic field. Increasing both the power and the magnetic field tend to increase n_e, which is not what the electrical analog model predicts in this case.

Some information on the discrepancies is gained from Figure 3-10. In this figure, the predicted value of n_e is plotted as a function of power for a magnetic field strength of 0 Gauss (black line, square symbols). This plot shows that the model predicts that the electron density decreases to a minimum at around 200 watts and increases after that. Also plotted on this graph is the optical emission signal for the Ar 7503.869 Å line as a function of power. The Ar emission strength should depend only on the concentration of

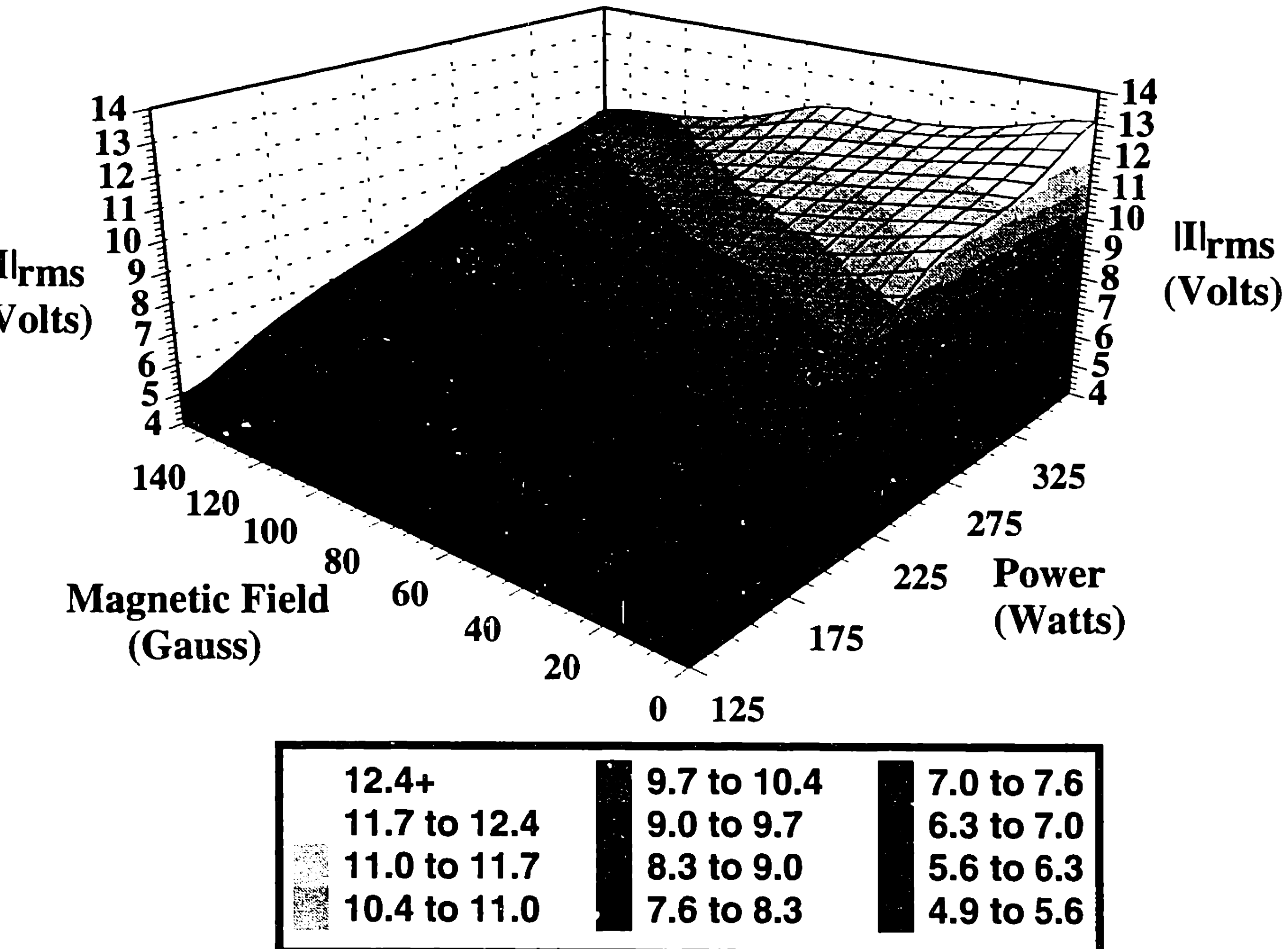


Figure 3-7: Modeled rms RF Current as a Function of Power and Magnetic Field Strength. Pressure = 100 mTorr, Composition = 25% HBr.

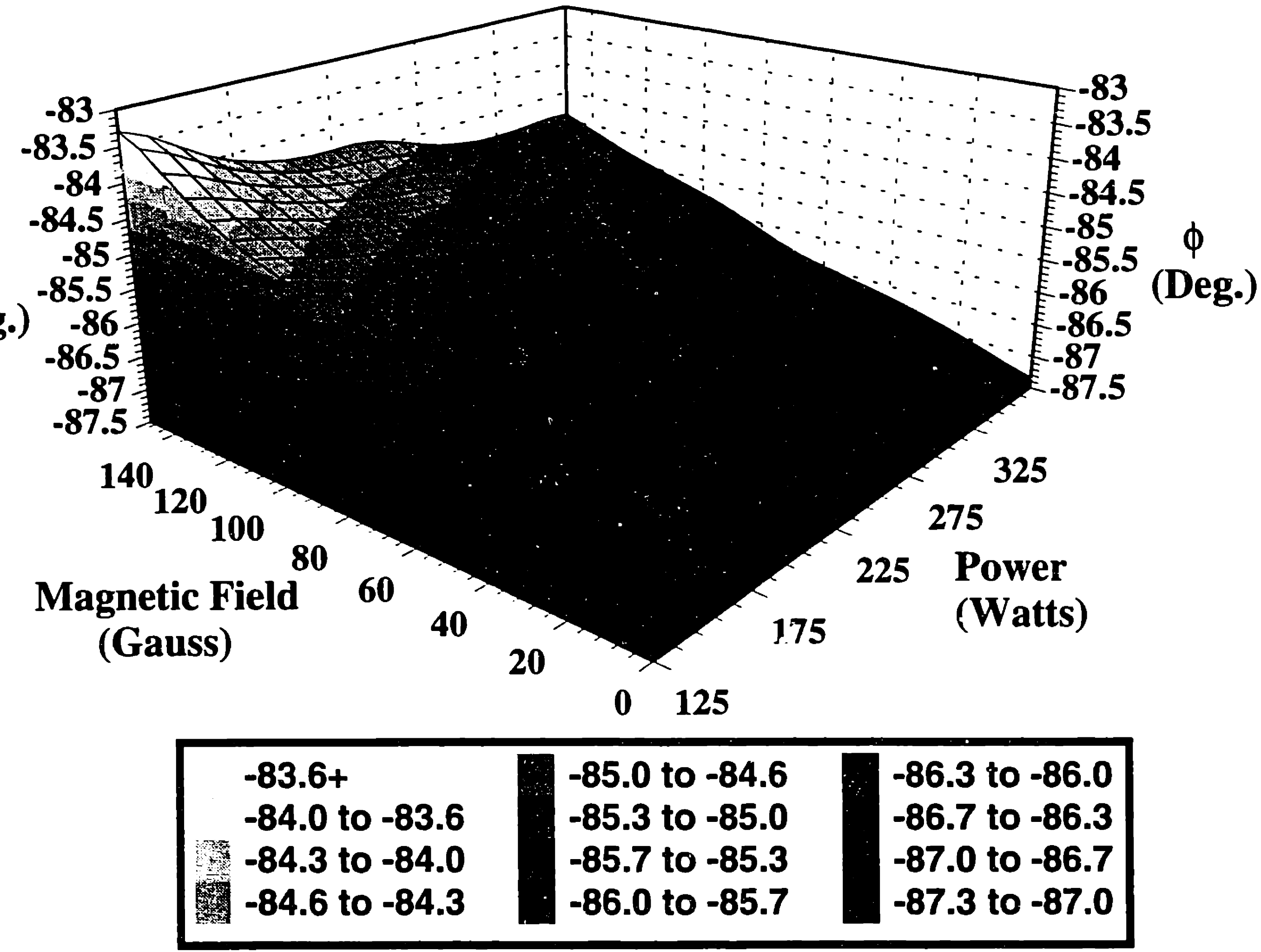


Figure 3-8: Modeled Phase Angle as a Function of Power and Magnetic Field Strength. Pressure = 100 mTorr, Composition = 25% HBr.

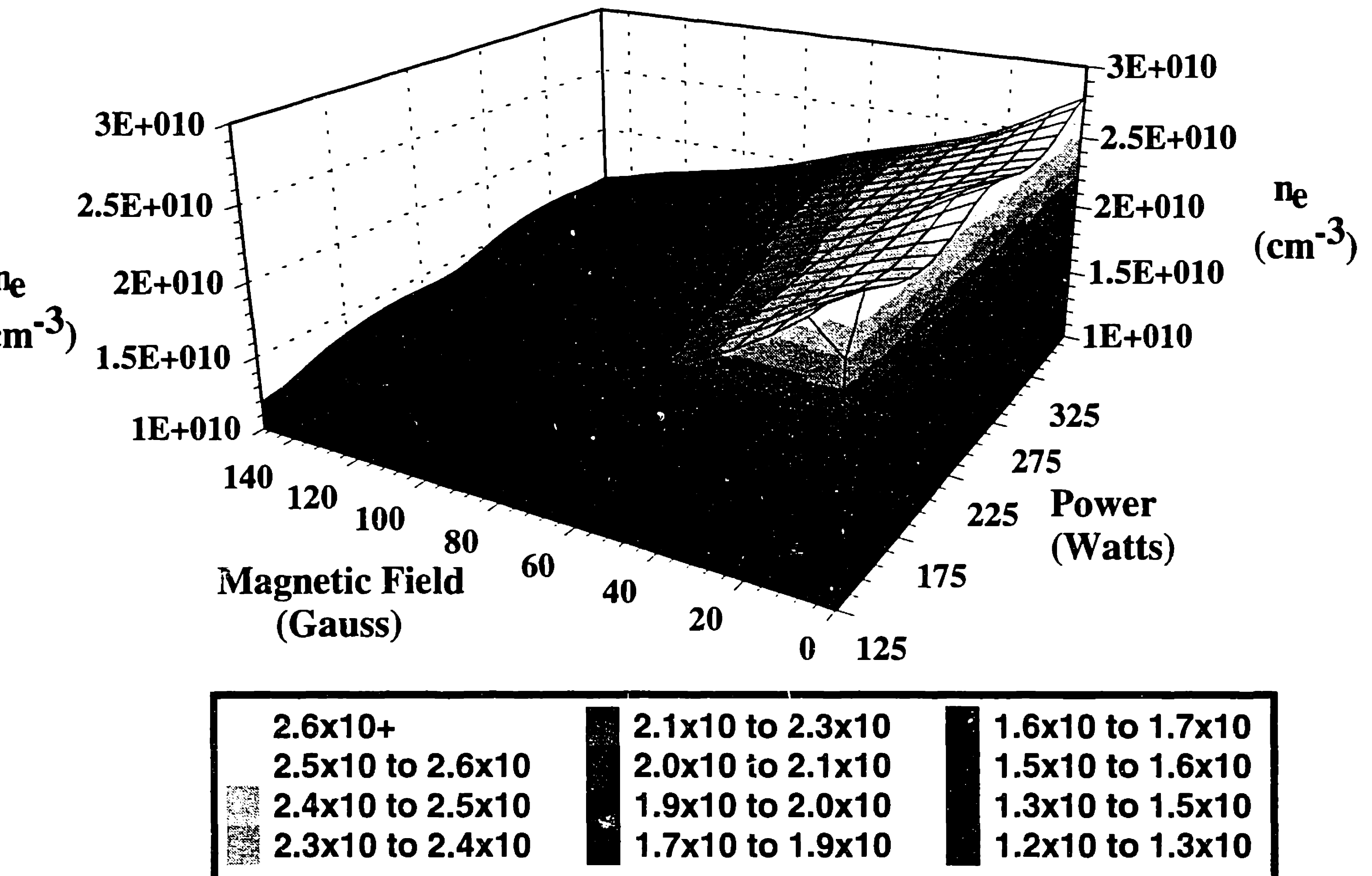


Figure 3-9: Predicted Electron Density as a Function of Power and Magnetic Field Strength. Pressure = 100 mTorr, Composition = 25% HBr.

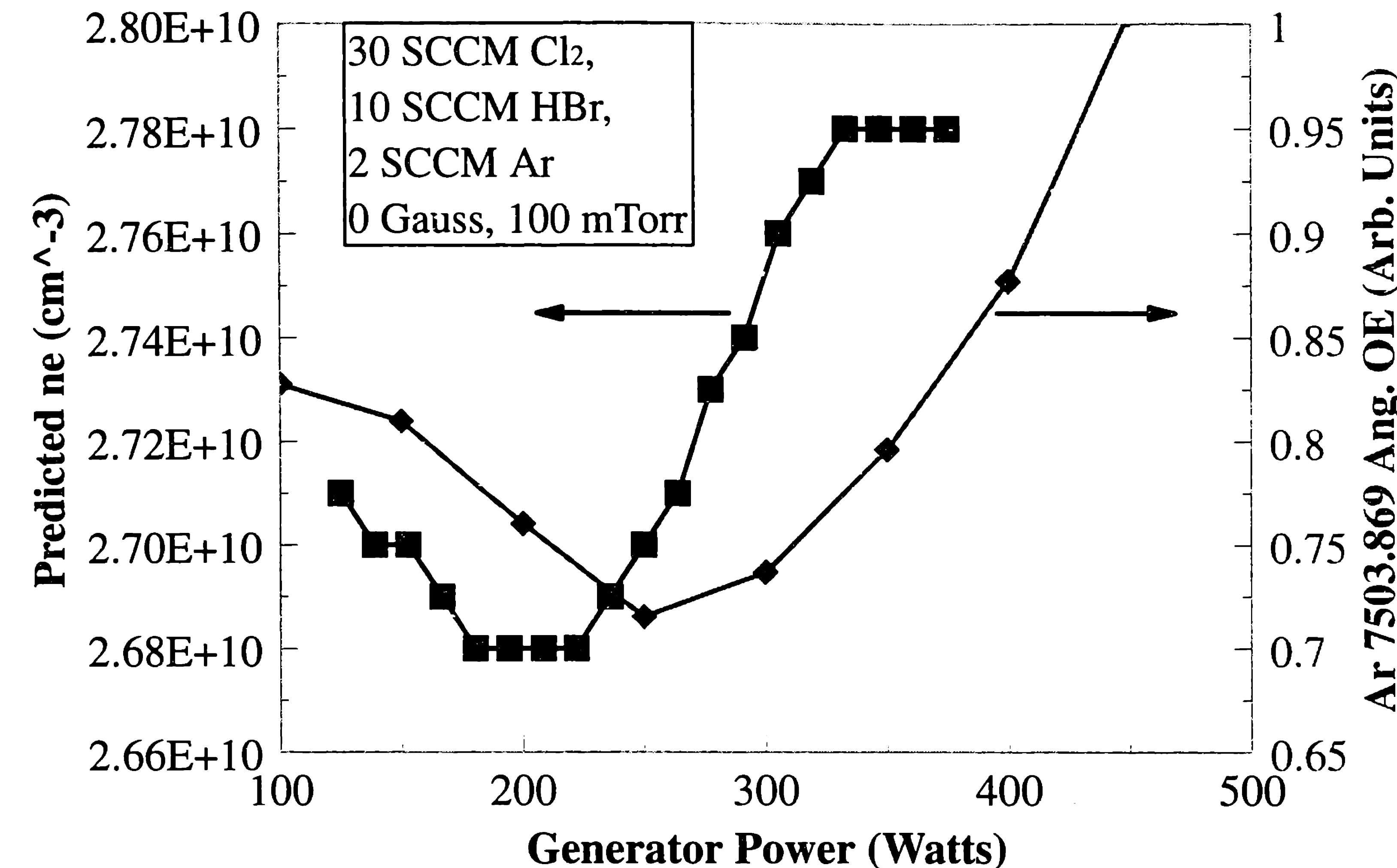


Figure 3-10: Comparison of Predicted Electron Density and Ar 7503.869 Å Optical Emission Intensity.

Ar (constant at 5%), and the electron density and electron temperature (Equation 2-2). This data shows a trend that is very similar to that predicted by the model. The minimum in AR emission occurs around 250 W. This difference in the location of the minimum could be due to: (1) an actual difference; (2) error in the model; or (3) drift of the generator calibration - there was an almost three year time span between the two data sets. Similar anomalous decreases in the electron temperature with increasing rf electric field have observed by others [Godyak and Piejak, 1993].

The correlation seen in Figure 3-10 for n_e and I_{Ar} as a function of power at B=0 was not obtained between n_e and I_{Ar} as a function of magnetic field strength at P=250 Watts (Figure 3-11). In this case, the neural network model predicts that n_e decreases with increasing field strength, while the Ar emission data shows that the emission intensity increases with increasing field strength (as expected). It is possible, but unlikely, that n_e is decreasing, but the electron temperature is increasing, leading to an increase in the emission intensity.

The apparent failure of the electrical analog model to predict the affects of magnetic field strength on n_e may be due to a number of reasons. First, the assumptions used to calculate n_e may not hold: (1) In this case, the electron density is not spatially uniform in the presence of a magnetic field due to magnetic field induced motion of the electrons as seen in the following equation for the force on an electron, **F**,

$$\boldsymbol{F} = e\boldsymbol{E}(t) + e\boldsymbol{V}\times\boldsymbol{B}(t) \qquad \textbf{(3-11)}$$

where **E**(t) is the electric field, **V** is the electron velocity, **B**(t) is the magnetic field (rotating), and e is the electronic charge; (2) Also, the plasma is not well confined to a fixed volume,

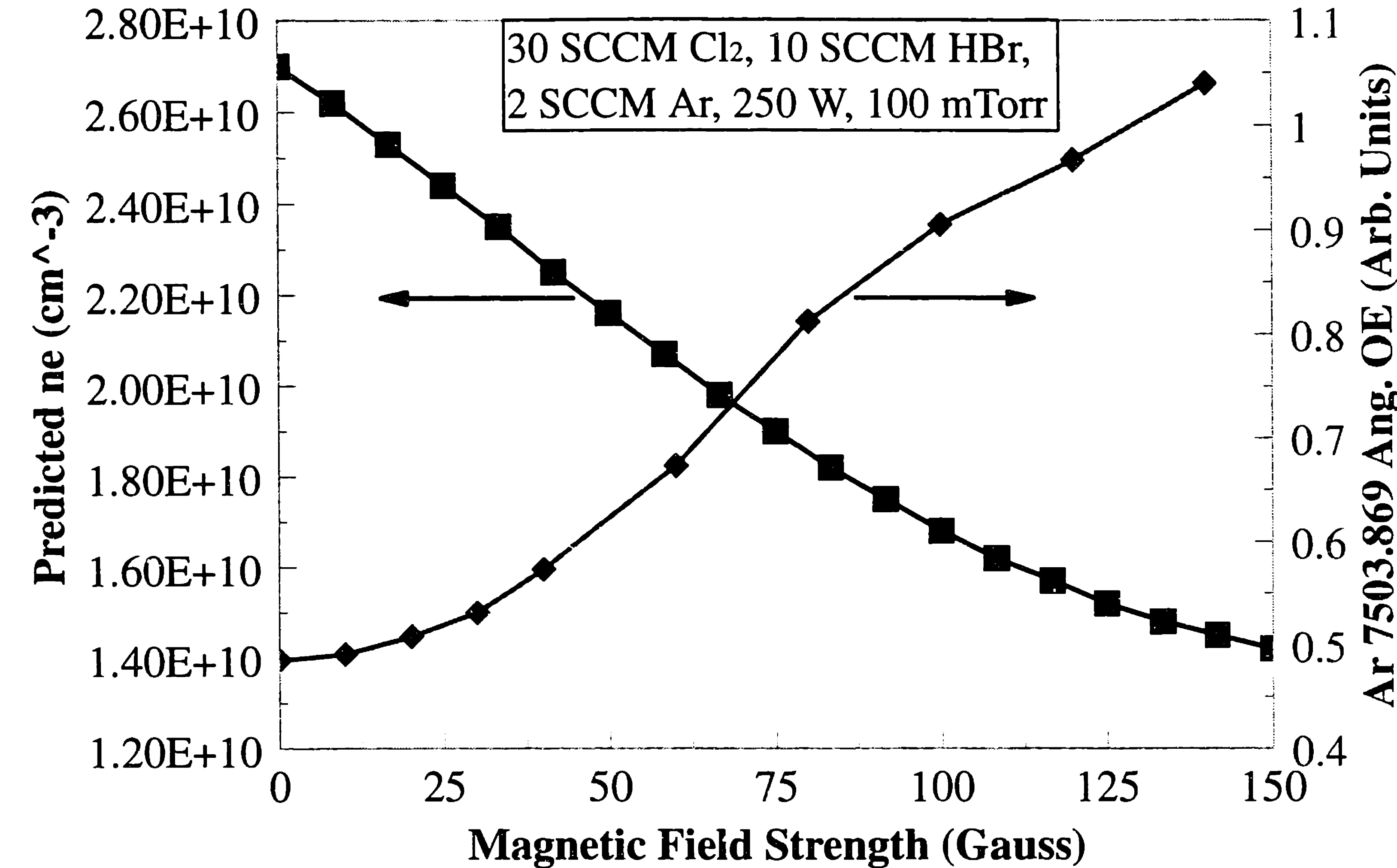


Figure 3-11: Comparison of Predicted Electron Density and Ar 7503.869 Å Optical Emission Intensity.

and instead may change size inside the etcher in response to inputs; (3) The plasma may be dissipating a significant amount of power in the sheaths and not in the bulk. Bulk power dissipation is expected at low pressures and high magnetic fields (as is the case here), but a switch over from bulk to sheath deposition is expected at high currents [Liu, 1993]; and (4) Significant amounts of current may be carried by species other than electrons, such as positive and negative ions.

The response of n_e to gas composition and pressure is shown in Figure 3-12 for the centerpoint conditions (power = 250 Watts, magnetic field = 75 Gauss). In this case, n_e is fairly insensitive to the gas composition. It does depend strongly on the pressure. As pressure decreases, n_e decrease; this is due to the decreasing gas density.

3.5 Etching Rate and Uniformity

The neural network model contained estimates for the etching rate at five locations on the wafer (left, center, right, top and bottom). From these five etching rates, an average etching rate was computed, along with an estimate of the etching rate nonuniformity. The values used to train and to test the neural network model are listed in Table 3-4, along with the calculated values of the average etching rate and the etching rate nonuniformity. The percent nonuniformity, NU, was calculated using

$$\mathrm{NU} = \frac{ER_{max} - ER_{min}}{<ER>} * 100\% * SIGN(m) \qquad \textbf{(3-12)}$$

where ERmax was the maximum etching rate of the 5 sites, ERmin was the minimum

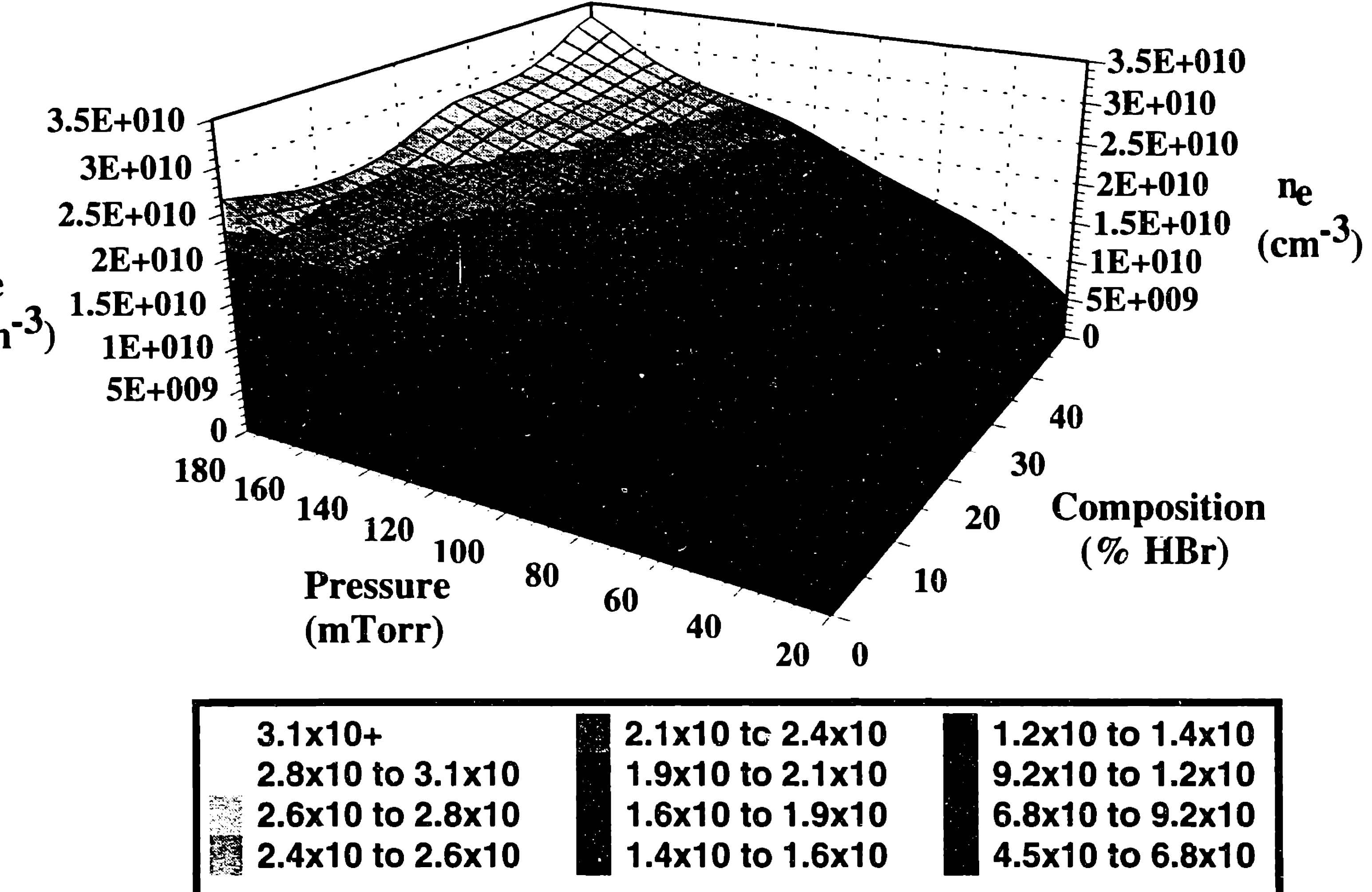


Figure 3-12: Predicted Electron Density as a Function of Composition (% HBr, Remainder Cl_2, 40 SCCM Total) and Pressure. P = 250 Watts, B = 75 Gauss.

Table 3-4

Etching Rates, Nonuniformity and Etchant Concentrations

Run	ER_{left} (Å/Min)	ER_{cntr} (Å/Min)	ER_{right} (Å/Min)	ER_{top} (Å/Min)	ER_{bot} (Å/Min)	<ER> (Å/Min)	NU (%)	[Br] (Arb.Units)	[Cl] (Arb.Units)
1	5248	5190	5216	5420	5413	5297	4.34	111	112
2	5212	5120	5111	5448	5386	5255	6.41	113	112
3	5248	5133	5116	5448	5404	5270	6.30	114	115
4	6907	6950	6812	6768	6978	6883	-3.05	0	237
5	5582	5757	5340	5350	5706	5547	-7.52	23.6	23.2
6	7322	7674	7187	7084	7553	7364	-8.01	133	135
7	3553	3248	3598	3577	3801	3555	15.55	157	116
8	5284	5834	5104	5194	5467	5377	-13.58	235.8	271.8
9	6701	7323	6485	6362	6730	6720	-14.30	136	155
10	2782	3119	2713	2723	2857	2839	-14.30	125	110
11	4260	4381	4216	4215	4267	4268	-3.89	236	18
12	5763	6340	5688	5567	5740	5820	-13.28	35.4	130.2
13	5475	5817	5301	5365	5769	5545	-9.31	133	143
14	3562	3717	3465	3453	3566	3553	-7.43	267.4	112
15	8468	8767	8440	8323	8687	8537	-5.20	0	163.2
16	5125	5272	4970	4949	5229	5109	-6.32	79.8	294
17	8124	8272	8073	8025	8262	8151	-3.03	77	285.6
18	7924	8129	7889	7640	7953	7907	-6.18	47.2	92.63
19	5331	5673	5175	5180	5582	5388	-9.24	125	126
20	6066	6007	6113	6049	6365	6120	5.85	33	108.6
21	4568	4373	4583	4548	4774	4569	8.78	117.6	37.8
22	4339	4850	4233	4193	4365	4396	-14.95	108	39
23	3389	3309	3362	3362	3515	3387	6.08	203.84	76.96
24	4810	4769	4736	4736	4974	4804	5.08	312.2	114.8
25	5607	5842	5436	5392	5823	5620	-8.01	124	129
26	4650	4635	4632	4632	4692	4648	1.29	82.6	285.6
27	3608	3351	3678	3607	3934	3636	16.04	19.04	47.6
28	2750	2765	2749	2738	2812	2763	-2.68	289.8	93.8
29	7358	7702	7358	7227	7237	7376	-6.44	113.4	42
30	4533	4617	4561	4519	4719	4590	-4.36	32.4	110.4
31	8318	8749	8280	7795	7888	8206	-11.63	39.37	34.41
32	5165	5616	5136	5013	5386	5263	-11.46	122	121
33	6319	6610	6092	6158	6190	6274	-8.26	257.6	109.2
34	2723	2947	2677	2627	2749	2745	-11.66	218.38	67.1
35	8146	8603	8144	7869	8111	8175	-8.98	31.8	108.6
36	7067	7222	6982	7033	7238	7108	-3.60	79.8	264.6
37	5047	5070	5057	4897	5187	5052	-5.74	26.88	62.58
38	5441	5781	5349	5350	5717	5528	-7.82	123	123
39	2789	2774	2783	2761	2888	2799	4.54	111.6	27

etching rate of the 5 sites, <ER> was the average etching rate, SIGN(x) was the sign function, yielding -1 if $x < 0$ and +1 if $x > 0$, and m was the radial slope of the etching rate variation. Thus, NU < 0 indicated that the etching rate was higher at center of the wafer than at the edge, while NU > 0 indicated that the etching rate was higher at the edge than at the center, with the numerical value of NU indicating the amount of variation across the wafer.

The average etching rate is plotted as a function of magnetic field strength and power in Figure 3-13, for the centerpoint conditions (pressure = 100 mTorr, composition = 25% HBr). The etching rate increases with increasing power and magnetic field, as expected. At 375 Watts, the etching rate increased by 67% (from 5100 to 8500 Å/min) as the magnetic field was increased from 0 to 150 Gauss. The increase in etching rate with power and magnetic field is due to the increased production of reactive species as these two inputs increase. Despite the decrease in mean ion energy as the magnetic field increases, the etching rate increases, indicated that the density of etching species is increasing, and that the reactor is probably operating in a neutral flux limited regime.

The nonuniformity of etching corresponding to the average etching rate shown in the previous figure is shown in Figure 3-14. Power has a very slight effect on the etching rate uniformity. Magnetic field has a very large effect on uniformity. Without a magnetic field, the etching rate is a minimum at the wafer center, and increases with increasing radial position (this is known as a "Bull's-eye" profile, and is discussed in more detail in Chapter 5); in this case, NU is ~ +10%. A low strength magnetic field decreases NU (at 250 W, NU is ~ +7% at 50 Gauss). Increasing the field strength further causes the profile of NU to switch drastically. At high magnetic fields, the etching rate is a maximum near the center and decreases with increasing radial position. Nu is ~ -7% at 75 Gauss, and becomes more

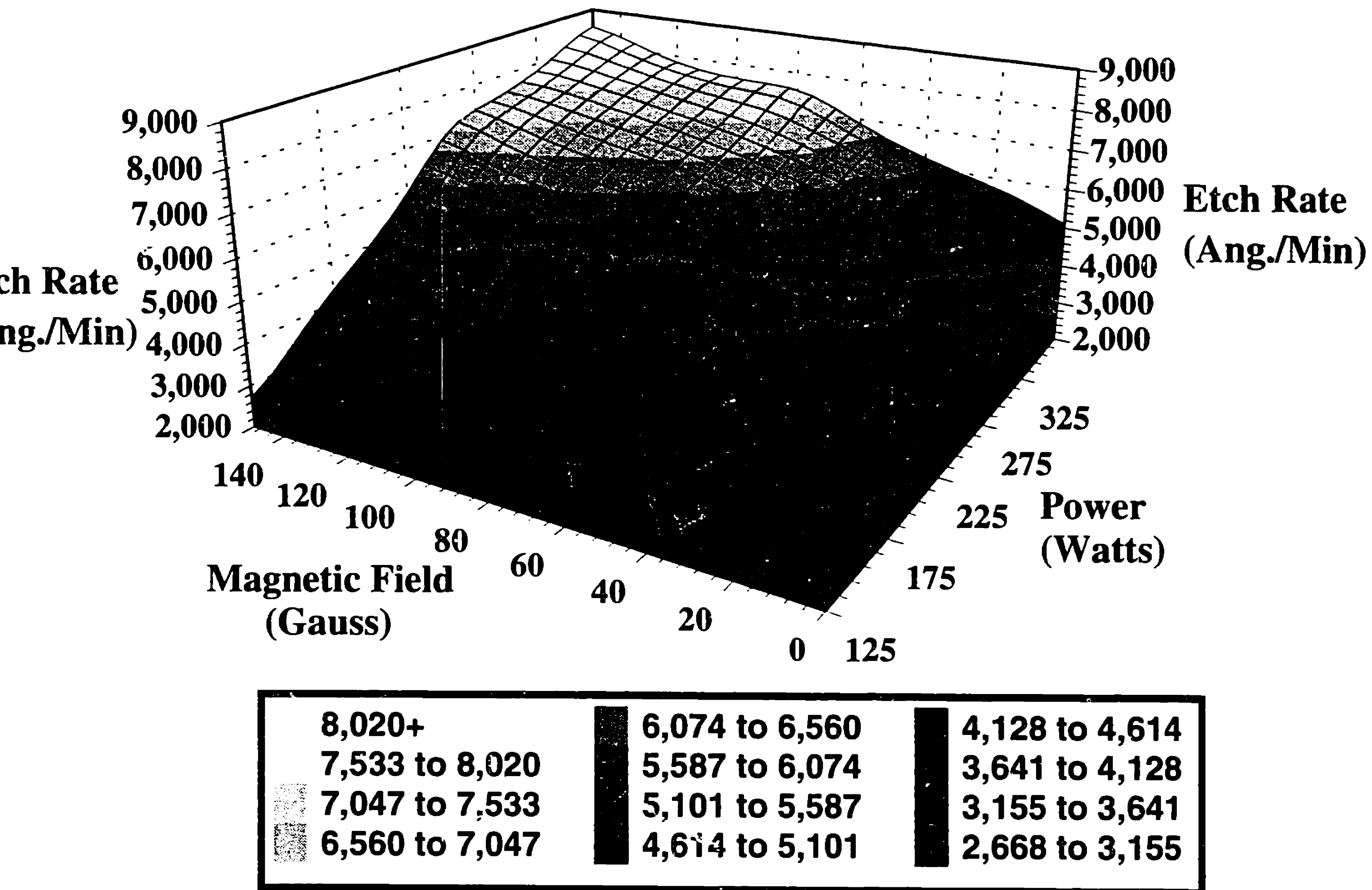


Figure 3-13: Average Etching Rate as a Function of Power and Magnetic Field Strength. Pressure = 100 mTorr, Composition = 25% HBr.

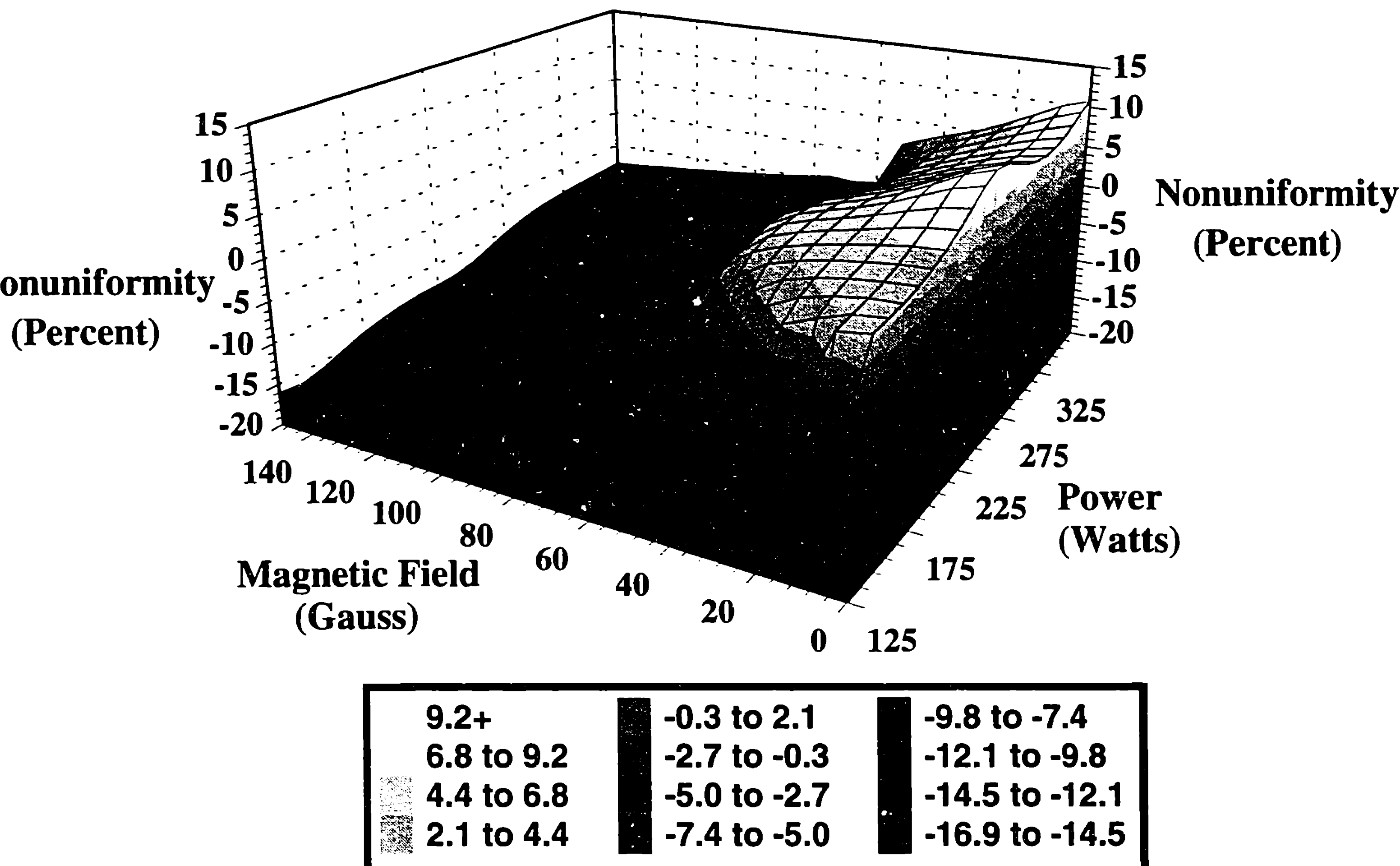


Figure 3-14: Etching Rate Non-Unifomrity as a Function of Power and Magnetic Field Strength. Pressure = 100 mTorr, Composition = 25% HBr.

negative with increasing field, becoming ~ -12% at 150 Gauss. The presence of a sharp plateau in the response of etching rate nonuniformity to magnetic field has been seen by others on the AME-5000 [Nguyen, 1990].

The presence of the magnetic field has a profound effect on the etching rate uniformity. Visually, the field changes the plasma uniformity. Without a magnetic field, the plasma glow appears to be of uniform intensity. However, in the presence of a magnetic field, the plasma glow is spatially nonuniform, indicating a spatially varying plasma density. The production of reactive species in the gas phase also varies spatially due to the density variations; this leads to etching rate variations across the wafer.

The etching rate variations indicate a competition between reactive species generation and diffusion, reaction, and convection out of the reactor. As explained in Chapter 8, the relative importance of diffusion and convection is expressed by the Peclet number, Pe (Equation 8-1). As seen in Figure 8-19, at 40 SCCM total flowrate, Pe calculated for a characteristic distance of 5 cm (wafer center to edge) is approximately 3.7, indicating that the loss of reactants by convection is as important as the diffusion of reactants across the wafer. The competition between diffusion of reactive species and the reaction rate is given by the Damköhler number, Da (Equation 8-2). A lower estimate for Da (Figure 8-20) from a plasma CSTR model presented in Chapter 8 yields $Da > 0.1$, and quite possibly $Da \sim 1$. This indicates that the reaction time is on the same order as the diffusion time.

The vales of Pe and Da indicate that reactive species generated in the plasma may be consumed or lost before having a chance to diffuse across the wafer to equalize distributions across the wafer. Thus, changing the location within the plasma where the etchant species are generated (with a magnetic field, for example) changes the etching rate nonuniformity.

The average etching rate as a function of chamber pressure and feed gas composition is shown in Figure 3-15. The average etching rate decreases with increasing HBr in the feed gas. This is due to the reactivity of HBr being less than that of Cl_2 (Table 1-1), so that as the HBr content increases, the average reactivity of the feed gas decreases. The etching rate decreases with increasing pressure, despite the fact that the total gas density increases, and the electron density increases (Figure 3-12). The drop is most likely due to the decrease in the ion energy caused by: (1) a decrease in the RF voltage driving the plasma (Figure 3-16), which corresponds to a drop in the RF sheath voltage (Equation 3-10); (2) a decrease in the DC bias (Figures 7-16, 7-17); and (3) a decrease in the mean ion energy caused by more gas ion-neutral collisions while crossing the sheath. This dependence of etching rate on pressure has been seen in Cl_2 discharges on an AME-5000 previously [Nguyen, 1990]. This behavior of the etching rate is indicative of a process that is highly ion-enhanced and not spontaneous.

The etching rate nonuniformity for the same conditions as those in Figure 3-15 is plotted in Figure 3-17. The feed gas composition has a slight effect on NU. Only at high pressure (180 mTorr) does the gas phase composition have a strong influence on NU. Similarly, pressure has only a slight effect on NU.

3.6 Cl and Br Concentrations

Ar was added to the plasmas used in this experimental design at a level of 7.5% (3 SCCM, using a 20 SCCM full-scale MFC) for use as an actinometer (Section 2.4.3). Optical emission from Ar (7503.869 Å) was used to normalize Cl (7547.869 Å) and Br (7005.19 Å) emissions to remove the effects of variations in the electron density and electron temperature

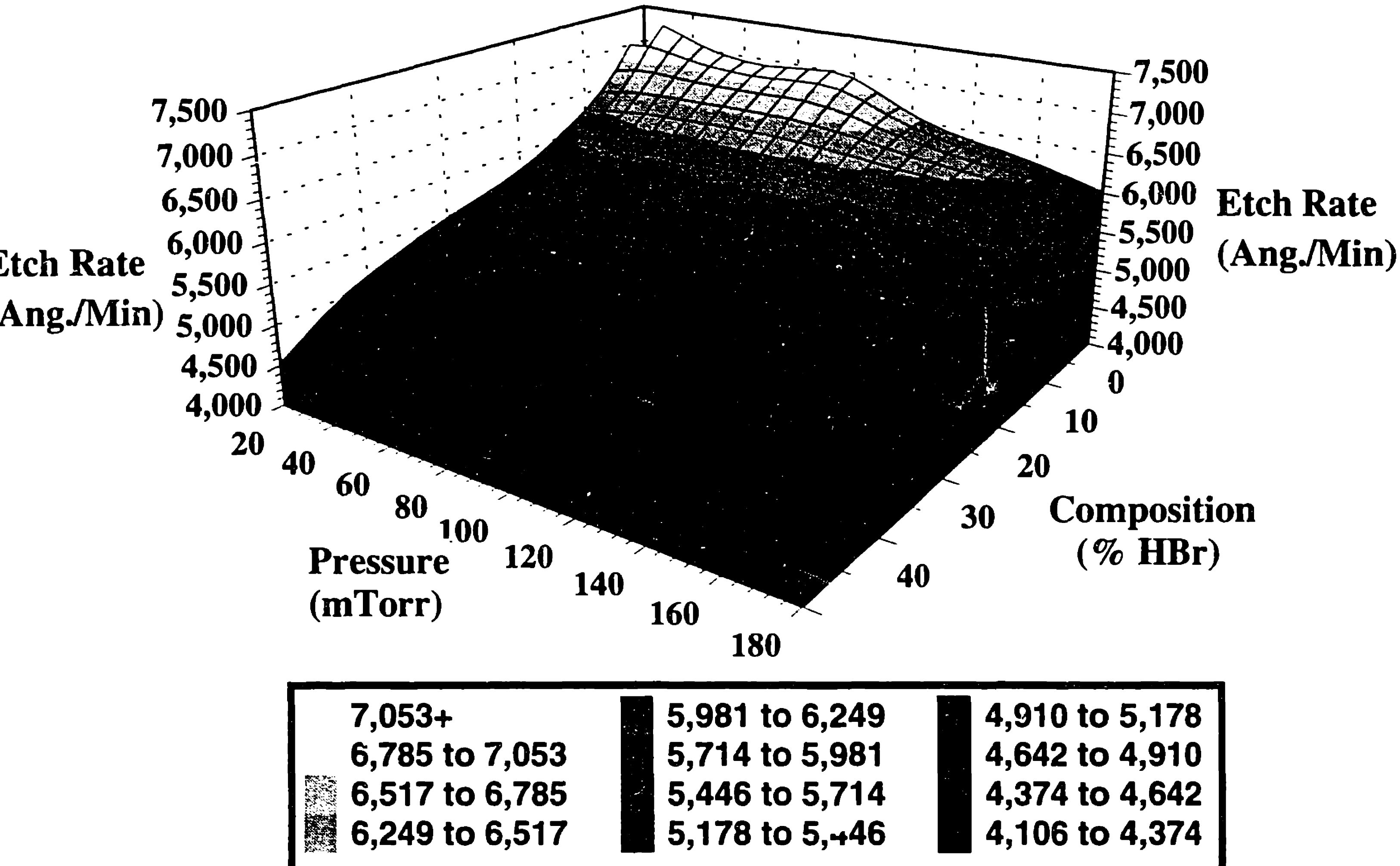


Figure 3-15: Average Etching Rate as a Function of Pressure and Feed Gas Composition (%HBr). Power = 250 W, Magnetic Field = 75 G.

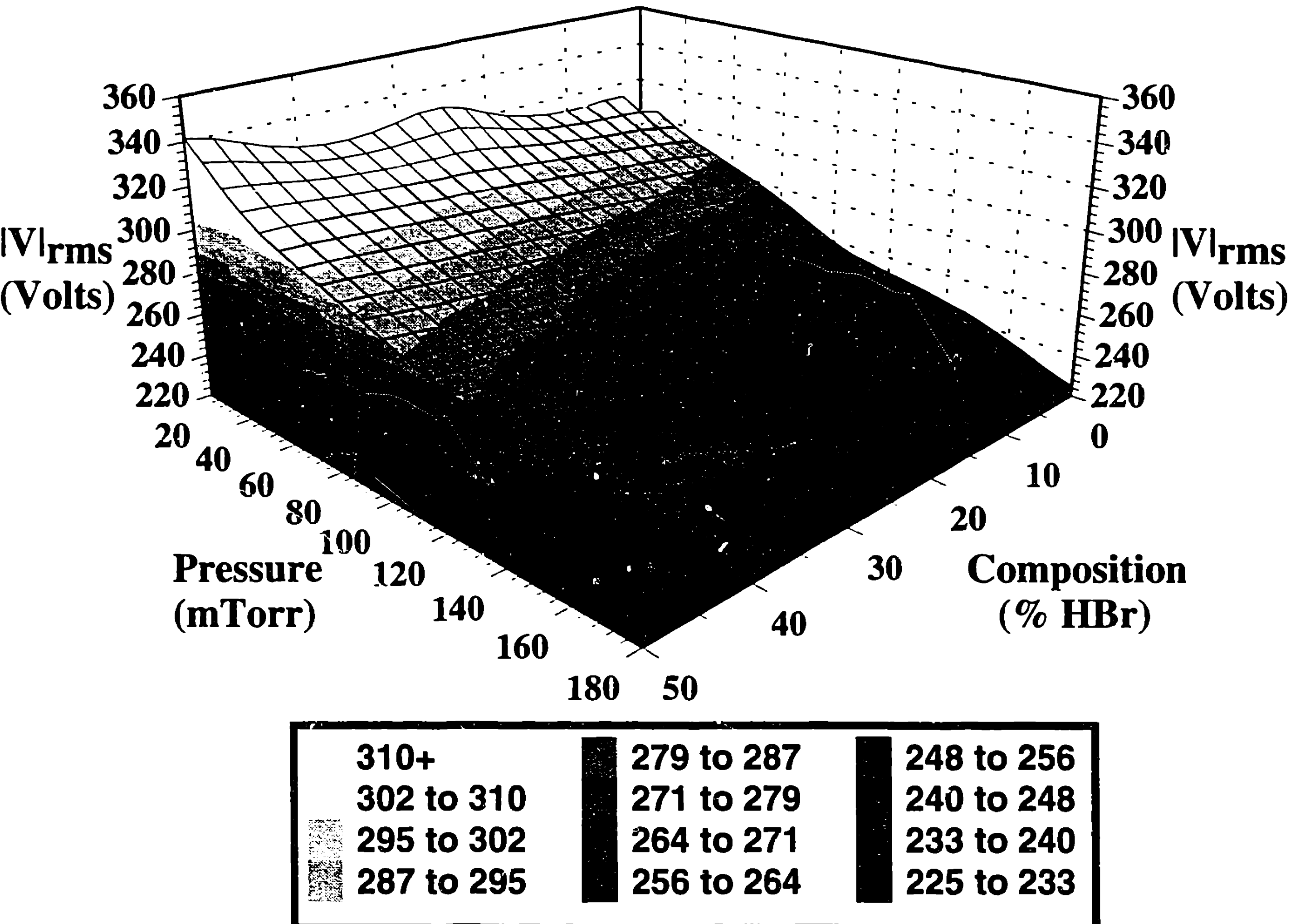


Figure 3-16: rms RF Voltage as a Function of Pressure and Feed Gas Composition (%HBr). Power = 250 W, Magnetic Field = 75 G.

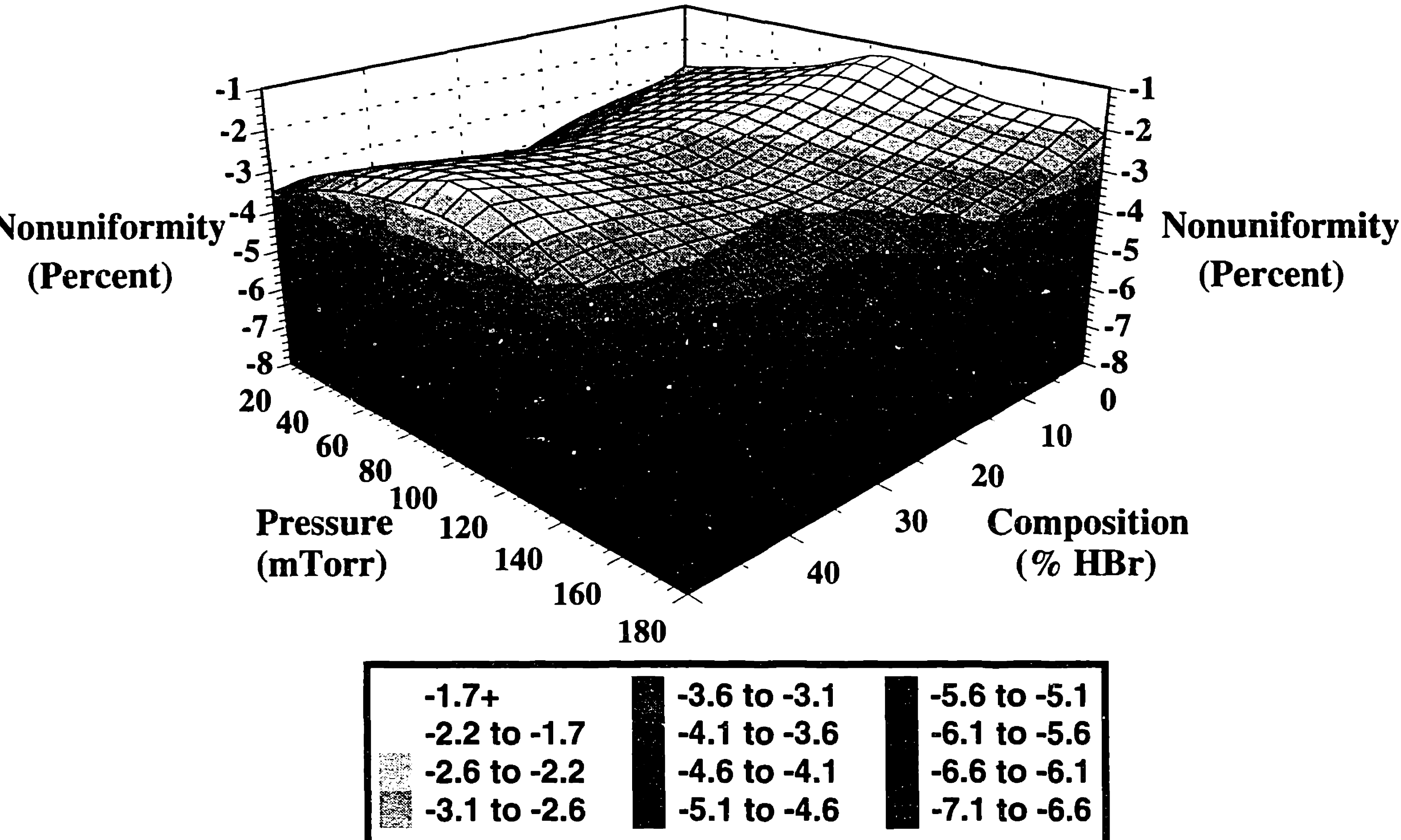


Figure 3-17: Etching Rate Non-Unifomrity as a Function of Pressure and Feed Gas Composition (%HBr). Power = 250 W. Magnetic Field = 75 G.

from the OE signals to yield an estimate of the concentrations of Cl and Br. Absolute concentrations ($\#/cm^{-3}$) could not be determined because of an unknown constant (K in Equation 2-6).

Br and Cl concentrations as functions of power and magnetic field are plotted in Figures 3-18 and 3-19, respectively, for a pressure of 100 mTorr and a composition of 25% HBr. Looking at the responses of both Br and Cl to power in the absence of a magnetic field, we observe the expected response. The concentrations of both of these species increase with increasing power. As more power is put into the reactor, more bonds can be broken, yielding a higher concentration of the Br and Cl.

The responses of Br and Cl concentration to magnetic field strength is quite different. Br decreases sharply with increasing magnetic field strength and then increase slightly again, while Cl increases. The different responses are due two things: (1) the effect of magnetic field strength on electron energy; and (2) the different bond dissociation energies of HBr and Cl_2.

The magnetic field changes the path that the electrons follow in the plasma (see Chapter 7, and Figure 7-1). The electrons gyrate around the magnetic field lines, effectively increasing their path in the plasma before they are lost at the chamber walls. This increased path increases the number of collisions an electron undergoes in the plasma, which raises the plasma density, but can decrease the electron temperature. The bond dissociation energies of HBr and Cl_2 are quite different. The Cl-Cl bond dissociation energy is 2.52 eV, while that of the H-Br bond is 3.80 eV [Weast, 1985].

The increase in Cl concentration with increasing magnetic field is due to an increase in the plasma density (both ion and electron densities). The response of Br to magnetic field

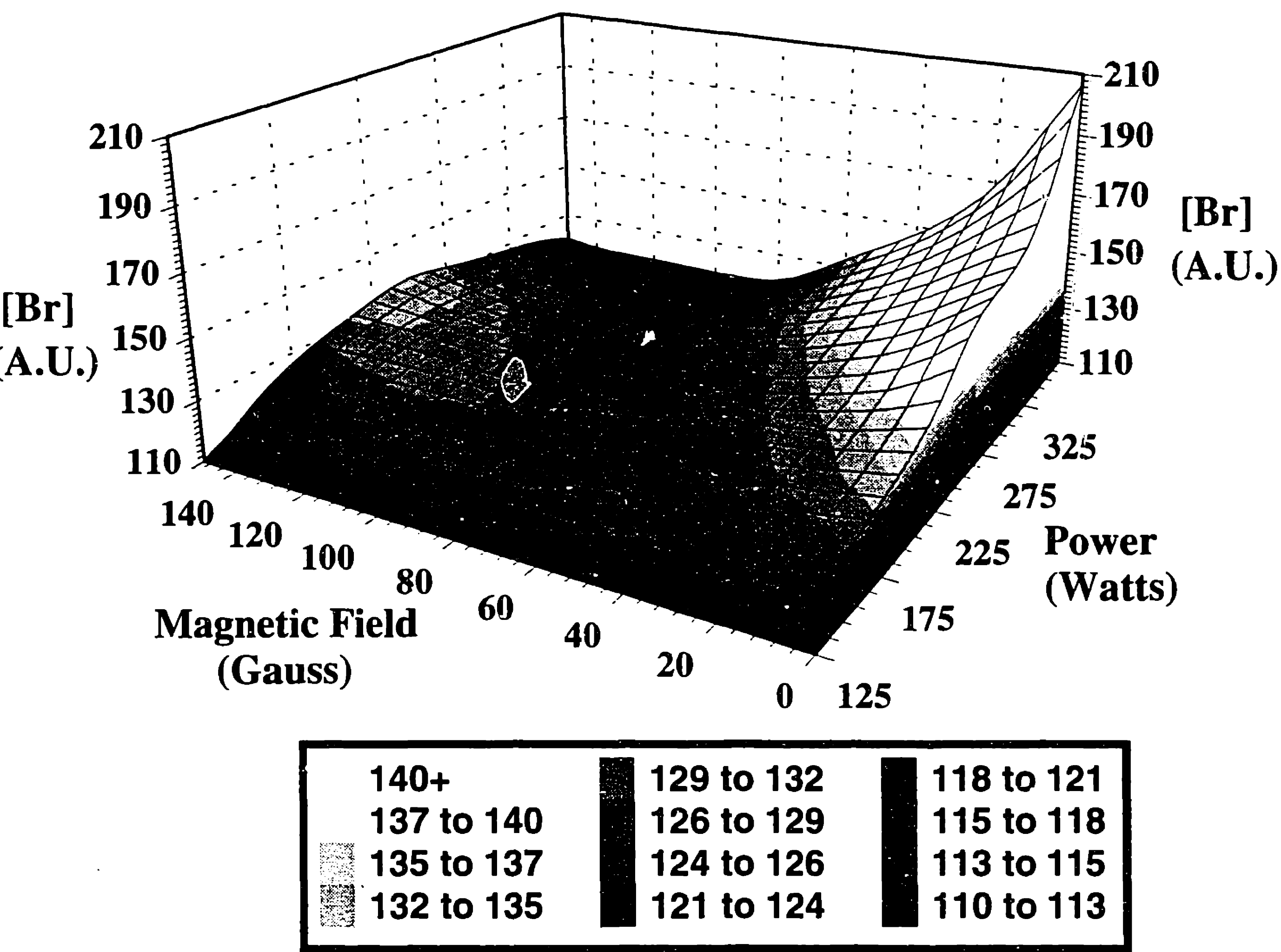


Figure 3-18: Modeled Bromine Concentration as a Function of Magnetic Field and Power. Pressure = 100 mT. Composition = 25% HBr.

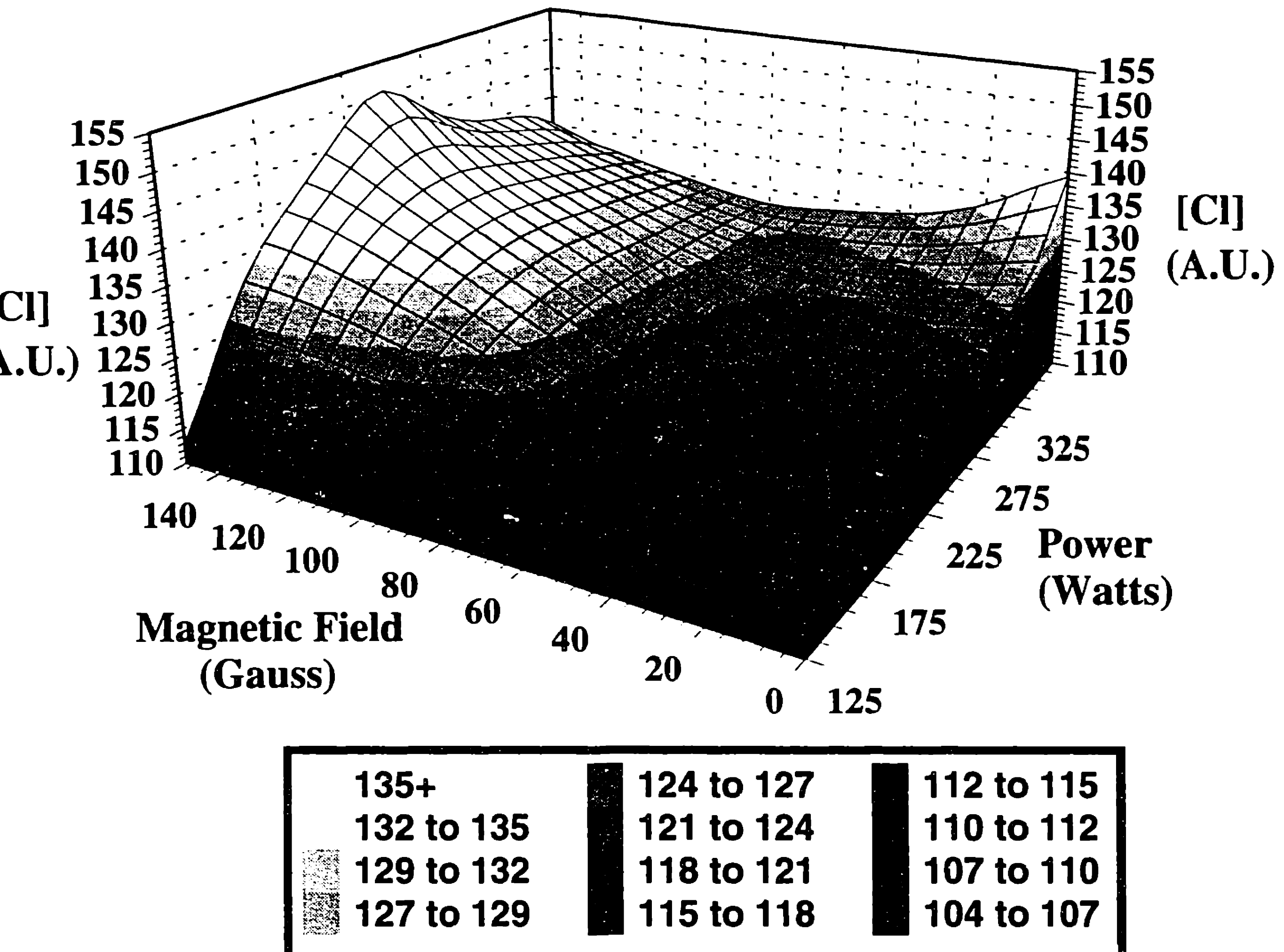


Figure 3-19: Modeled Chlorine Concentration as a Function of Magnetic Field and Power. Pressure = 100 mT. Composition = 25% HBr.

strength involves a competition between a decrease in T_e and an increase in plasma density. The initial decrease of Br concentration with increasing magnetic field is due to decreasing electron temperature, T_e. If T_e is close to the H-Br dissociation energy, and if increasing the magnetic field slightly decreases T_e, then the production of Br by electron impact dissociation of HBr will decrease. The drop in T_e will not affect Cl because the Cl-Cl bond dissociation energy is 1.28 eV lower. The slight increase of Br with magnetic field strength at higher fields is due to the increase in the plasma density more then offsetting the decrease in T_e.

The decrease in T_e with increasing magnetic field strength is due to the increased confinement of the electrons; confining the electrons to motion along the magnetic field lines leads to an increase in plasma density and to an increase in electron impact dissociation of Cl_2. However, the increased number of electron-neutral collisions can lower T_e, despite the electrons longer residence time within the accelerating electric field.

The response of Br to the magnetic field strength for pressures of 20, 100 and 180 mTorr is shown in Figure 3-20. The Br concentration is normalized for each pressure to the Br concentration without a magnetic field. All three show Br decreasing with increasing pressure, with Br increasing again after about 100 Gauss at 100 and 20 mTorr. The magnitude of the decrease is greatest at the lowest pressure. This agrees with the notion that the magnetic field is changing *Te* because the plasma is "more magnetized" at lower pressures. As the pressure decreases, the electron mean free path for electron-neutral collisions, λ_e increases, while the electron Larmor radius, r_l, remains fairly constant (see Chapter 7).

The responses of Br and Cl to pressure and feed gas composition at 250 Watts and 75 Gauss are shown in Figures 3-21 and 3-22, respectively. Both plots show that Br and Cl

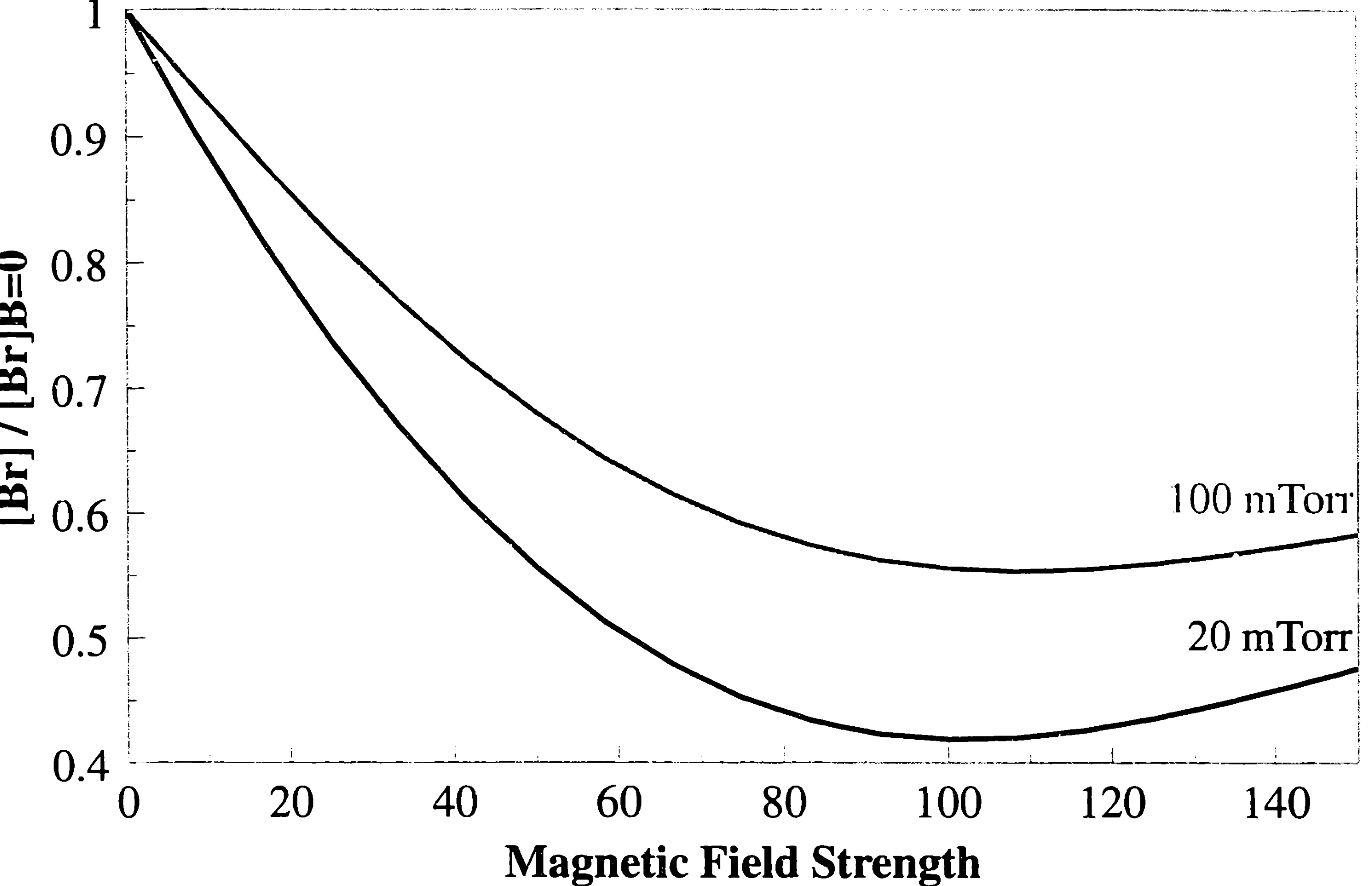


Figure 3-20: Effect of Magnetic Field on [Br]. Concntrations Normalized to Concentration Without Magnetic Field. (375 Gauss, 25% HBr).

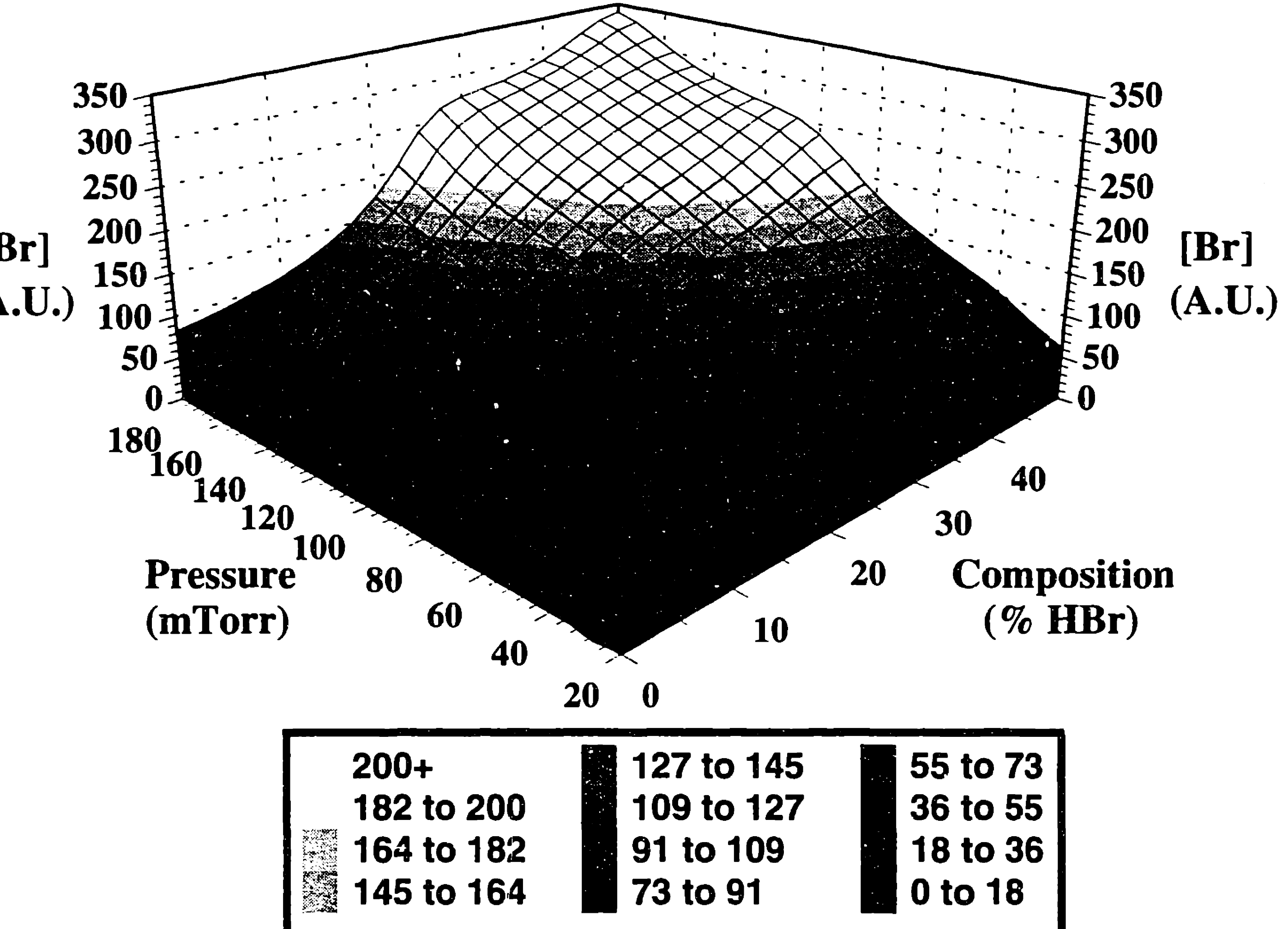


Figure 3-21: Modeled Bromine Concentration as a Function of Composition (% HBr, Remainder Cl_2, 40 SCCM Total) and Pressure. P = 250 Watts, B = 75 Gauss.

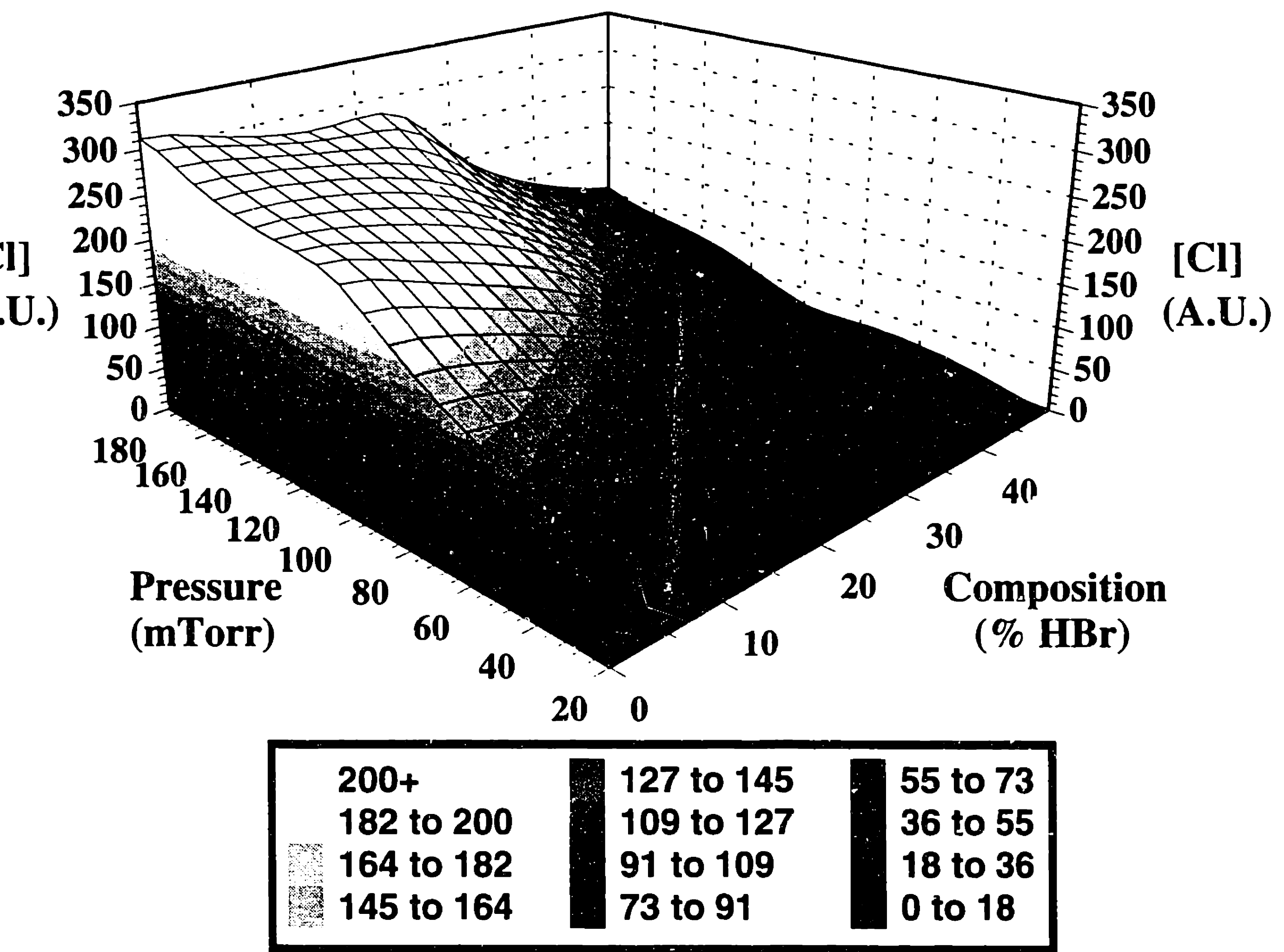


Figure 3-22: Modeled Chlorine Concentration as a Function of Composition (% HBr, Remainder Cl_2, 40 SCCM Total) and Pressure. P = 250 Watts, B = 75 Gauss.

both increase with increasing pressure. This is due to the increased total density of the gas with increasing pressure. Also, as expected, Br increases with an increasing amount of HBr in the feed gas, while Cl increases with a decreasing amount of HBr in the feed gas.

3.7 Conclusions

A neural network was used to construct an experimental design used to characterize polysilicon etching on an AME-5000. The inputs in the design were: power, pressure, magnetic field strength, and feed gas composition (percent HBr), at 40 SCCM total flowrate of Cl_2 + HBr. The measured outputs were: etching rate at 5 sites on the wafer, the rms RF current, voltage, and phase angle between them, and Br and Cl concentrations. The etching rates were determined by measuring film thickness before and after a partial etch with a Nanospec AFT/200. The electrical properties were measured using a Comdel RPM-1. The Br and Cl concentrations were measured using Ar actinometry, with 7.5% Ar added to the feed gas.

A total of 39 etching experiments were conducted using 5000 Å of undoped polysilicon patterned with KTI-820 photoresist. The neural network was unable to fit output variables that were calculated from other output variables, such as plasma impedance, plasma power and etching rate uniformity, therefor, these variables were not used as outputs. Instead, the raw data was used and these quantities were calculated from that data.

A 4 input, 10 hidden node, 10 output neural network model was found to best fit the data. Increasing the hidden nodes beyond 10 resulted in overfitting the model to noise in the data, while decreasing the hidden nodes below 10 resulted to too simple a model that was not

able to fit complex curvatures in the data.

An electrical analog model that represents plasma impedance as a resistor for the bulk and a capacitor for the sheaths, was constructed, and used to calculate such plasma properties as electron density and electric field to pressure ratios. The predicted electron densities showed seemingly anomalous effects of power and magnetic field. However, the decrease in electron density with increasing power was verified with Ar optical emission intensity. The decrease in electron density with magnetic field was an incorrect prediction of the model due to the breakdown of one or more of the assumptions used to make the model.

The etching rate uniformity was strongly dependent on the magnetic field. Without a field, the etching rate increased with radial position. With a magnetic field, there was a sharp transition to the etching rate decreasing with increasing radial position. The decrease in etching rate with pressure indicate that polysilicon etching was strongly ion-enhanced.

The Br and Cl concentrations showed different responses to magnetic field. Cl increased with increasing magnetic field, due to the increased plasma density. Br decreased with increasing magnetic field, and in some case increased again at high fields. The Br response can be understood to be a competition between two processes. First, the magnetic field increases electron confinement, increasing the number of collisions they undergo, lowering the electron temperature. And second, the magnetic field increases the plasma density. Initially, the decrease in electron temperature results in a decrease in Br. However, at higher magnetic fields, the increase in electron density offsets the decrease in electron temperature, yielding an increase in Br. The decrease in Br is more severe at lower pressures, where the plasma is more magnetized. Cl does not undergo this behavior because the bond dissociation energy of Cl-Cl is 1.28 eV lower than that of H-Br.

Chapter 4

Microtrench Formation During Plasma Etching

The formation of microtrenches in polysilicon plasma etching over thin gate oxides has been observed and modeled. Microtrenches are small trenches formed in the bottom of etching features and are caused by the localized breakthrough of the gate oxide and subsequent rapid etching of the underlying silicon. In contrast to previous reports, the microtrenches were observed a small distance away from the sidewall. Their formation and position were functions of the thickness and angle of the photoresist and polysilicon sidewalls. Simulation of ion scattering from the sidewalls of the etching features indicated that the flux of ions at the bottom of the feature was peaked away from the sidewall under the process conditions of this study. The position of the peak ion flux predicted by the model and the experimentally observed trench varied in a similar fashion as a function of the topography of the etched feature.

4.1 Introduction

Microtrenching was observed in the plasma etching of submicron trenches in polysilicon after overetching. We define a microtrench as a narrow groove etched in the bottom of a nominally rectangular feature. The film structure being etched in this study consists of photoresist patterned polysilicon over a thin gate oxide on the silicon substrate. The microtrenches are formed when the etching process removes all of the

oxide from a small area and "punches through" to the silicon substrate below. The etching chemistry employed is highly selective to silicon, so upon breakthrough, the silicon substrate etches rapidly resulting in deep microtrenches. A Scanning Electron Micrograph (SEM) is shown in Figure 4-1 of a typical trench with microtrenches clearly visible. In this and all other SEM images is this paper, the photoresist has been removed prior to imaging.

Microtrenches and other notches have been observed on both feature bottoms and sidewalls during etching and ion-beam milling by other researchers. In ion-beam milling, a beam of energetic ions is used to physically etch a surface. It has been observed that trenches are formed on feature bottoms at the base of photoresist sidewalls during ion-beam milling [Glöersen, 1975],[Lee, 1979]. Simple ion reflection models can explain the observed trenches. [Younger and Haynes, 1982].

Microtrenching has been observed on feature bottoms during the etching of polysilicon over oxide with Cl_2 [Nguyen *et al.*, 1991], in Si etching with Cl_2/BCl_3 [Ameen et al., 1988], and in Si etching with Cl_2, Cl_2/H_2, and $Cl_2/SiCl_4$ [Sato and Arita, 1987]. Notches have also been observed in sidewalls [Ameen *et al.*, 1988], [Ohki et al., 1987]. Although not specifically addressed, microtrenches are visible in other systems such as etching of SiGe/Si in HBr [Bestwick et al., 1991].

Microtrenches reported by other researchers *e.g.* [Nguyen *et al.*, 1991] have typically been found immediately adjacent to the sidewall. The microtrenches observed in this experiment are unusual in that they are located several hundred angstroms from the sidewall. In this work, the frequency and severity of microtrenching varied with feature depth and aspect ratio (depth/width). Samples in which the resist displayed slight

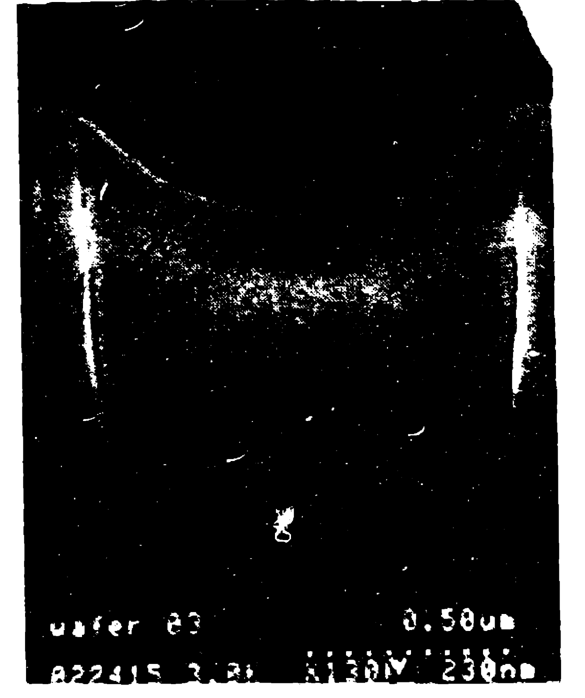


Figure 4-1: SEM Photograph of a Typical Line-Space Pattern, Exhibiting Microtrenches. Microtrenching Occurs During a Highly Selective Overetch, in Which the Thin Gate Oxide is Breached, and the Underlying Silicon is Rapidly Etched.

asymmetry resulted in microtrenching near one sidewall but not the other.

With the thickness of gate oxide decreasing, it is becoming more important to understand the cause of microtrench formation. Modifications can be made to the process or to the wafer topography to minimize microtrenching and ensure device integrity only after the underlying physics of microtrench formation are understood. This chapter first reviews the experimental procedures and observations. Next, a numerical model of the etching process is described which simulates the effect of ion scattering from the feature sidewalls. Finally, the simulation and experimental results are compared and shown to be consistent leading us to conclude that microtrenching is caused by ion scattering within a feature.

4.2 Experimental Procedures and Results

All samples were undoped polysilicon (≈ 3000 Å) over a thin gate oxide (less than 100 Å). Masking patterns were formed using positive photoresist and a stepper. The resist thickness was varied between 0.5 to 1.1 μm. The nominal linewidth between 0.35 and 0.5 μm were studied. After lithography, the wafers were etched using a two step process in a single-wafer parallel plate rf etching reactor. A rapid initial etch was performed to remove the bulk of the polysilicon film. This was followed by a more selective $Cl_2/O_2/He$ overetch at approximately 400 mTorr. An example of the film structure at polysilicon endpoint is shown in Figure 4-2. The polysilicon sidewall had a slope of 70-85° (depending upon process conditions), and the resist had a slope of 80-90°. The wafers were subjected to a substantial (greater than 80 percent) overetch to

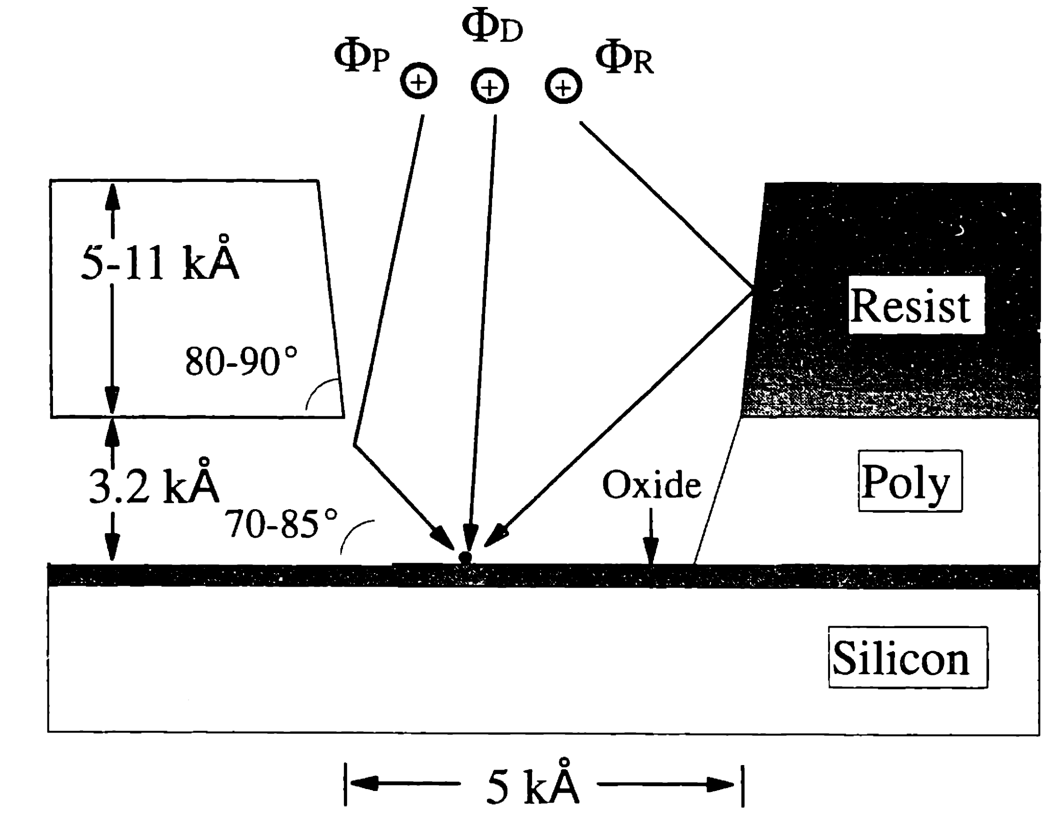


Figure 4-2: Cross Section of the Film Structure at Polysilicon Etching End-Point, Showing Typical Values for Polysilicon and Photoresist Sidewall Angles. Also Shown are the Three Components of The Ion Flux Arriving at a Point on the Feature Bottom.

ensure the removal of all polysilicon residues from the feature bottoms. It was during this overetch that the microtrenches formed. No trenching of the polysilicon was observable before or upon clearing of the polysilicon.

A matrix of resist thicknesses and line spacings was run, and the following trends observed:

Resist thickness - microtrenching was more frequent and more severe with thicker resist.

Aspect ratio - microtrenching was more common at higher aspect ratios (photoresist plus polysilicon thickness/space width). However, the overall correlation of microtrenching to aspect ratio was weak; two features with different resist thicknesses but the same aspect ratio would show different microtrenching behavior.

Asymmetry - the asymmetry (microtrenching on one side of a feature but not the other) was qualitatively correlated to asymmetry in the resist angle.

Location - the distance between the microtrenches and the sidewall was relatively constant. The microtrench width was also stable, so the microtrenches began to overlap at very narrow linewidths.

The resist angle could not be controlled independently of the photoresist thickness, making the resist thickness the only adjustable process parameter. Under these

conditions, an aspect ratio of about 3 was identified as a critical condition, with microtrenching common for greater aspect ratios.

4.3 Model Formulation

4.3.1 Possible Causes of Microtrenches

A theoretical model for microtrenching was constructed to explore the cause of microtrenching in the system described above. A search of the literature yielded the following three possible mechanisms for etching non-uniformity along the feature bottoms:

Surface Diffusion [Sato and Arita, 1987]- surface diffusion of neutral reactants along the sidewall, where the ion flux and reactant consumption are low, to the bottom of the feature. The additional flux of etchant from the sidewalls leads to a greater reactant concentration at the sidewall bottom corner. The higher ion flux at the bottom combined with the excess reactant concentration at the corner leads to the formation of trenches at the corners where the concentration is highest. This theory could not account for the observed microtrench location which is away from the corner.

Ion Reflection [Nguyen *et al.*, 1991]- specular reflection of glancing ions from the

sidewalls to the feature bottom. Ions striking the sidewall at near-grazing incidence, which do not embed and transfer their energy to the wall, are specularly reflected and strike the bottom. Ions arriving at higher incidence angles embed in the wall losing most of their energy in the collision. The effect on bottom etching rate depends upon the feature geometry, the probability of reflection (as a function of incidence angle), and the angular distribution of ions arriving from the plasma (the Ion Angular Distribution, or IAD). The expected trends were conceptually consistent with the experimental observations, but verification of this possibility required numerical simulation. Ion reflection is thought to be responsible for other etching phenomena as well. Oehrlein *et al.* [1990] concluded that ion reflection from oxide mask sidewalls prevented the formation of black silicon in a region adjacent to the mask when etching silicon with $HCl/O_2/BCl_3$.

Surface Charging [Arnold and Sawin, 1991]- localized charging of insulating surfaces (primarily the resist) which causes the formation of electric fields which deflect ion trajectory, thereby focusing the ions at the bottom of the feature. The charging has been theorized to be caused by the convergence of the directional ion flux and the isotropic electron flux to form an intermediately directed flux in which all positions along the feature surface receive a net neutral flux. This phenomenon would not focus the ions to a single region along the feature bottom, because its primary effect is increase the ion flux to the sidewall. However, surface charging could contribute to ion reflection by increasing the number of sidewall collisions.

4.3.2 Profile Modeling

Of the three possible causes for microtrench formation listed above, ion reflection was judged to be most plausible, and a numerical simulation of the surface profile during etching was constructed. The simulation utilized an existing profile evolution code. [Jackson and Dalton, 1989],[ASEPEN, 1990]. This program has been discussed elsewhere, but a few salient features will be reviewed here. The surface is represented by a "string" of straight line segments. Each segment is advanced in time along its inwardly pointing normal. This algorithm yields more accurate behavior at sharp corners than some codes which move points along the perpendicular bisector of the lines formed with the adjacent points.

The rate of advancement for each segment is calculated as proportional to the incident ion flux where the proportionality constant is a function of the material. This may be thought of as a "constant yield" reaction model, where the number of surface atoms liberated per incident ion is independent of ion energy and angle. Neutral reactants are assumed to be present in sufficient concentration to be approximately uniform throughout the feature. For example, a segment near the bottom of the feature may be advanced more slowly because the geometrical shadowing reduces the incident ion flux. Spontaneous (i.e. isotropic or chemical) etching mechanisms are not considered in this work.

4.3.3 Ion Flux Calculation

The addition of ion scattering to our profile evolution model changes the computation of ion flux to each segment. The total flux arriving at a point on the bottom is made up of three components as shown in Figure 4-2. The first is the direct flux (ions arriving directly from the plasma), also known as the "background" flux because this is the only flux present when ion reflection is not considered. The other two contributions are ions reflected from the polysilicon surface and from the resist surface. Only single scattering events are considered in this model. The inclusion of multiple scattering for this feature geometry would not significantly change the results as the first scattering event typically results in a high impact angle collision with the second surface. The etching rate and ion flux were computed as

$$\begin{aligned} ER &\propto \Phi_T , \\ \Phi_T &= \Phi_D + \Phi_P + \Phi_R , \end{aligned} \qquad (4\text{-}1)$$

where ER is the etching rate; Φ is the ion flux, with the subscripts T,D,P,R standing for the total flux, the flux coming directly from the plasma without collision, the flux reflected from the polysilicon, and the flux reflected from the resist, respectively.

The computation of the flux coming from the plasma without a collision is shown schematically in Figure 4-3a. A surface segment is characterized by two limiting view angles, α_L and α_R. These view angles are shown along with an IAD in Figure 4-3b. These view angles, along with surface orientation with respect to the ion angular distribution, are used to calculate the direct flux to a small surface segment as given by

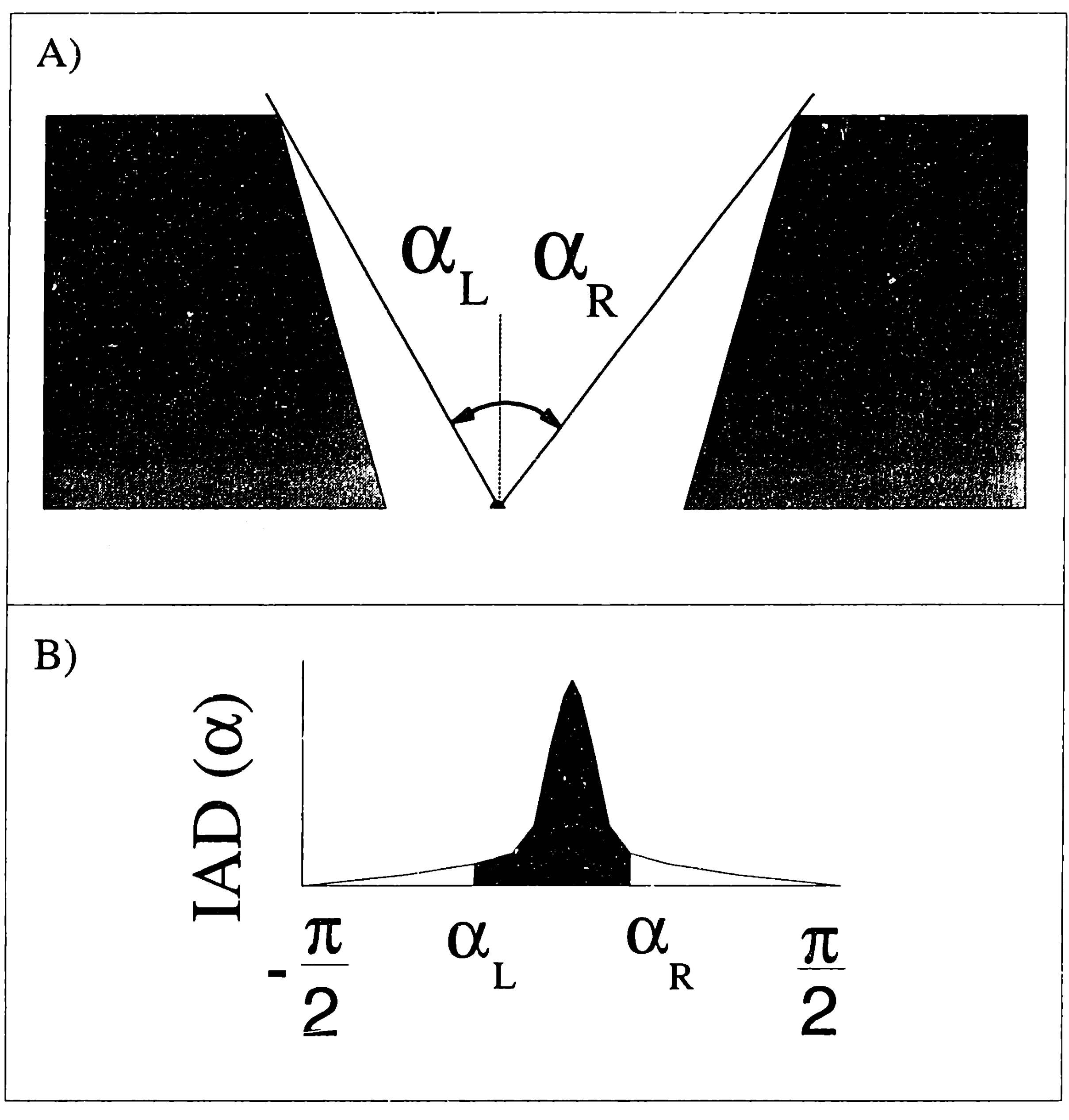


Figure 4-3: Definition of View ANgles and the Integration of the Ion Flux to the Feature Bottom Between the View Angles. (A) Nomenclatures for Calculation of Direct Flux, Φ_D. α_L and α_R are the "left" and "right" Limits of the Plasma View. (B) The Integration of a Typical Ion Angular Distribution to Determine the Ion Flux.

$$\Phi_D = \int_{\alpha_R}^{\alpha_L} IAD(\alpha)\ \cos(\alpha)\ d\alpha . \qquad (4\text{-}2)$$

Calculation of the reflected flux is more complicated, as shown in Figure 4-4. $P(\theta_i)$ is the probability of specular reflection for the incidence angle θ_i. With this term added, the general form for Φ_P or Φ_R becomes

$$\Phi_{P,R} = \int IAD(\theta_g)\ P(\theta_i)\ \cos(\alpha)\ d\alpha , \qquad (4\text{-}3)$$

where θ_i is the ion incident angle on the sidewall, θ_g is the ion angle in the gas phase, and α is the ion angle at the surface. Equations can be written for unknown angles in terms of known angles,

$$\begin{aligned} \theta_g &= 2\theta_s + \alpha - \pi , \\ \theta_i &= \theta_s + \alpha - \frac{\pi}{2} , \end{aligned} \qquad (4\text{-}4)$$

where θ_s is the sidewall angle. The limits of integration for calculation of Φ_P and Φ_R are determined by the view angles of the polysilicon and photoresist surfaces.

4.3.4 2-D vs. 3-D Calculations

The above equations apply to the two-dimensional case, while ion transport and reflection are inherently three-dimensional processes. However, the two-dimensional description may be used in some cases, e.g. a long symmetric trench. The two-dimensional solution is an accurate description if reflection from the sidewalls does not

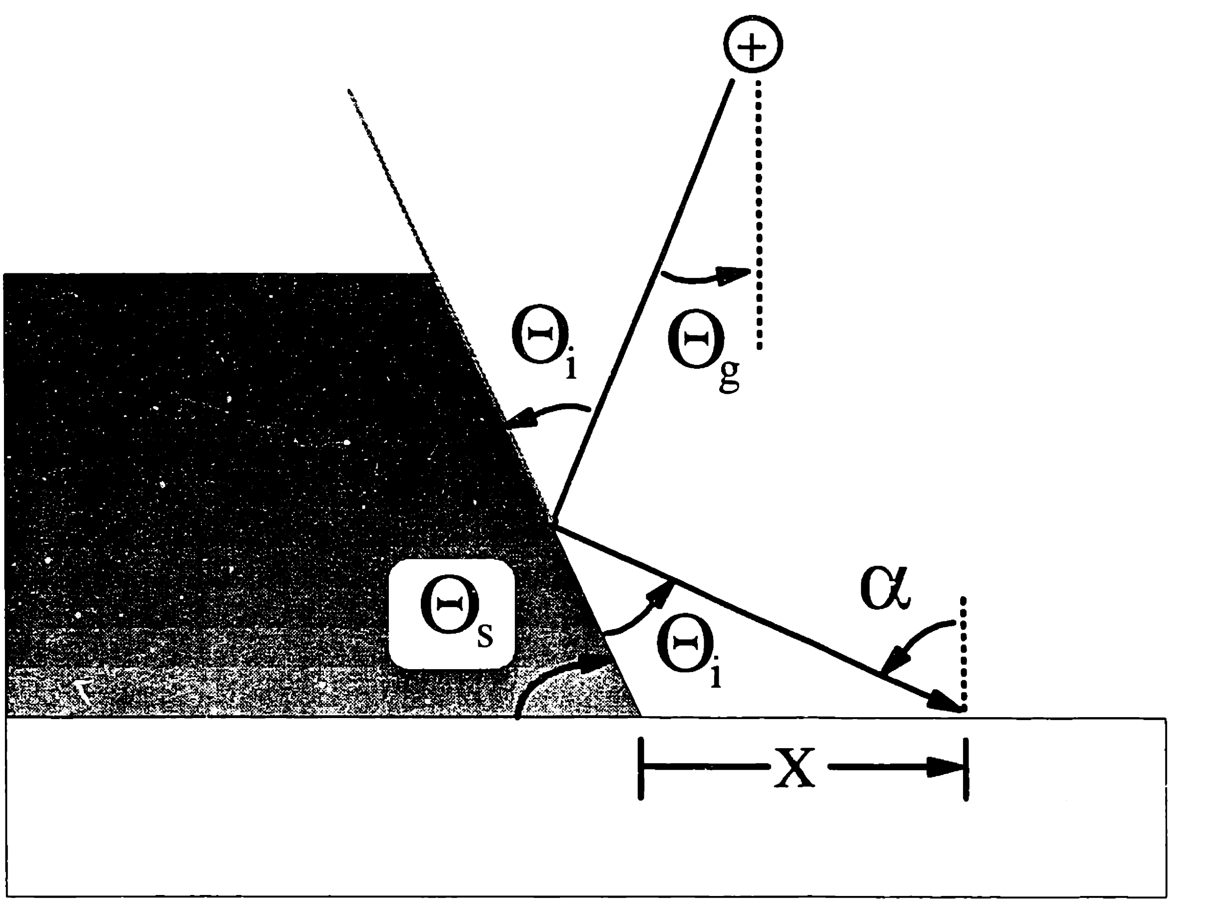


Figure 4-4: Nomenclature for Calculation of Reflected Fluxes, Φ_P or Φ_R; α is the Surface Arrival Angle, θ_s is the Inclanation Angle of the Sidewall at the Point of Reflection, θ_i is the Incidence Angle Upon the Sidewall, θ_g is Ion Angle from the Gas Phase.

affect the longitudinal (along the trench) component of ion velocity as in elastic scattering. In this case, the trajectory of the two-dimensional case is identical to the two-dimensional projection of the three-dimensional case; this is shown in Figure 4-5. All ions approaching a vertical sidewall with a given V_x and V_y appear equivalent, regardless of V_z.

The use of a two-dimensional model, however, requires the computation of appropriate weighting factors for the incident ion angular distribution to account for the collapse of the three-dimensional IAD into a two-dimensional IAD. The appropriate weighting is computed by integrating the three-dimension IAD over the out-of-plane angle. This leaves only one error in the reflected flux profile, arising from the use of a two-dimensional θ_i rather than the correct three-dimensional incident angle when evaluating the reflection probability. Comparison of the true three-dimensional computation with the above two-dimensional approximations yielded an error of less than 5% over the range of sidewall angles considered. Therefore, the 30 times increase in computational cost for the full simulation was not justified.

4.3.5 Ion Angular Distribution

The ion angular distribution was assumed to be that of an argon discharge at similar pressures. Ions suffer collisions with neutral molecules during their transit across the sheath. These collisions scatter the ions, giving them a transverse velocity component and an off-normal impingement angle. As a result, significant numbers of ions strike the feature sidewalls. Scattering within the sheath is particularly pronounced at higher

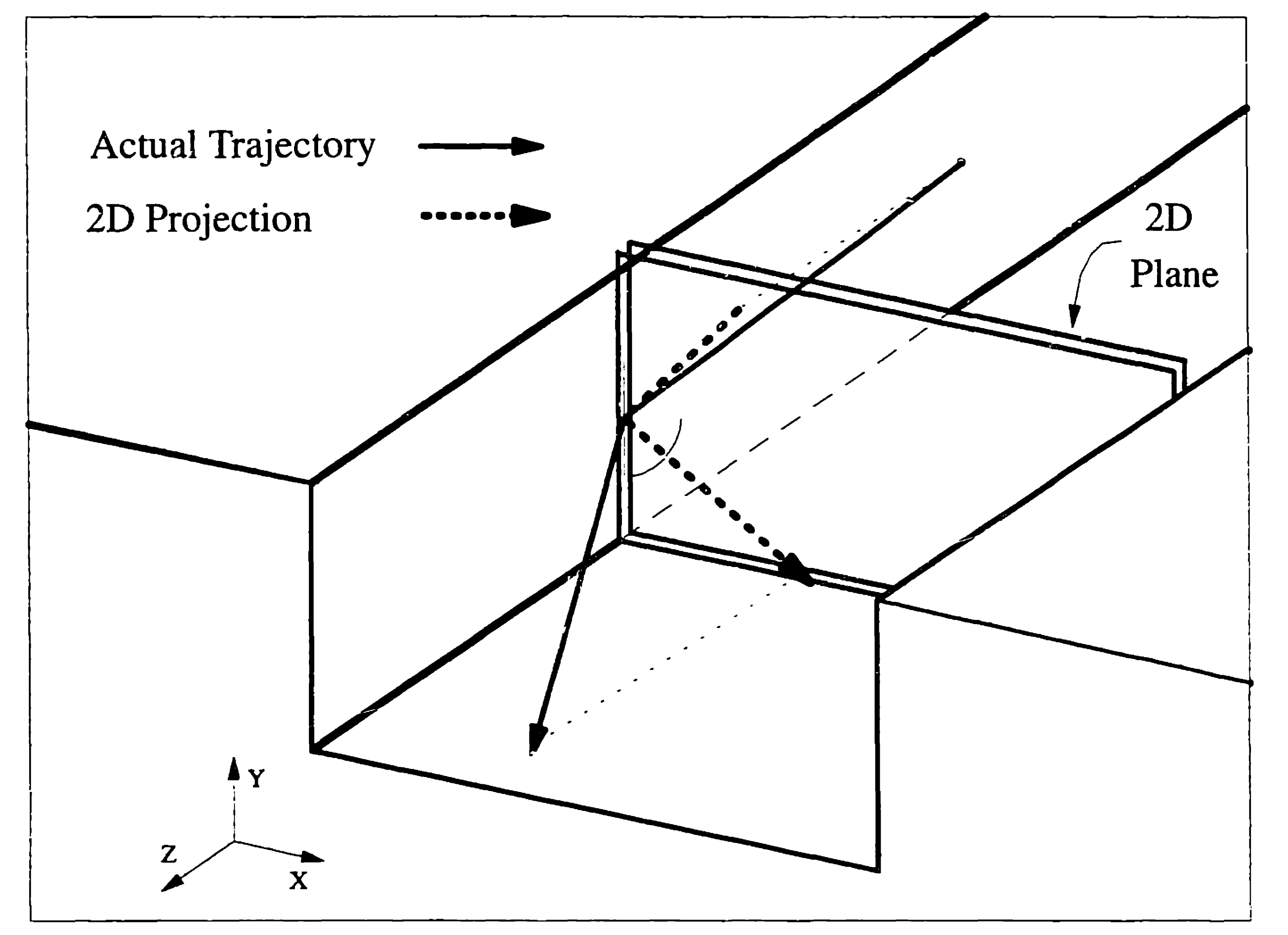


Figure 4-5: Equivalence of Two-Dimensional and Thre-Dimensional Trajectories for Ions Undergoing Specular Reflection From Sidewall. An Ion Following the Actual Trajectory (Solid Line) Appears to Follow the 2-D Trajectory (Dashed Line) when Viewed from the Longitudinal Trench Axis.

pressures (where the sheath is more collisional, resulting in an increased number of ion collisions during ion transit of the sheath), such as the 400 mTorr used in these experiments. The authors are aware of no published data concerning the IAD for Cl^+, therefore, the published data of Liu *et al.* [1990] for Ar^+ in the pressure range of interest was used. Their data for 500 mTorr was used as a starting point. This IAD and its two-dimensional weighted distribution are shown in Figure 4-6.

The use of an experimental Ar^+ IAD as an approximation for the Cl^+ distribution is valid. There is evidence that in a Cl_2 discharge, Cl_2^+ may be the dominant chlorine ion, with densities two to three orders of magnitude greater than Cl^+ [Donnely and Flamm, 1985]. Cl_2^+ has a significantly higher charge-exchange collision cross section with Cl_2 than does Cl^+ because of symmetry. [Hasted, 1979],[Stebbings *et al.*, 1963]. Thus, Cl_2^+ is expected to behave in a manner similar to Ar^+ in which charge-exchange collisions are important in determining the IAD and the Ion Energy Distribution. [Liu *et al.*, 1990],[Kushner, 1985]. For the sake of simplicity, we will hereafter refer to chlorine ions present in the plasma simply as Cl^+.

The probability of ions being reflected from a surface rather than inducing a surface reaction, sputtering the surface, and/or imbedding in the surface is not well known. The reflection of ions from surfaces has been studied previously [Barrett, 1971],[Biersack and Haggmark, 1980], but the reported results are typically for higher ion energies and different materials. Data for < 1 KeV Cl^+ ions and silicon surfaces could not be found in the literature.

4.3.6 Ion Specular Reflection Probability

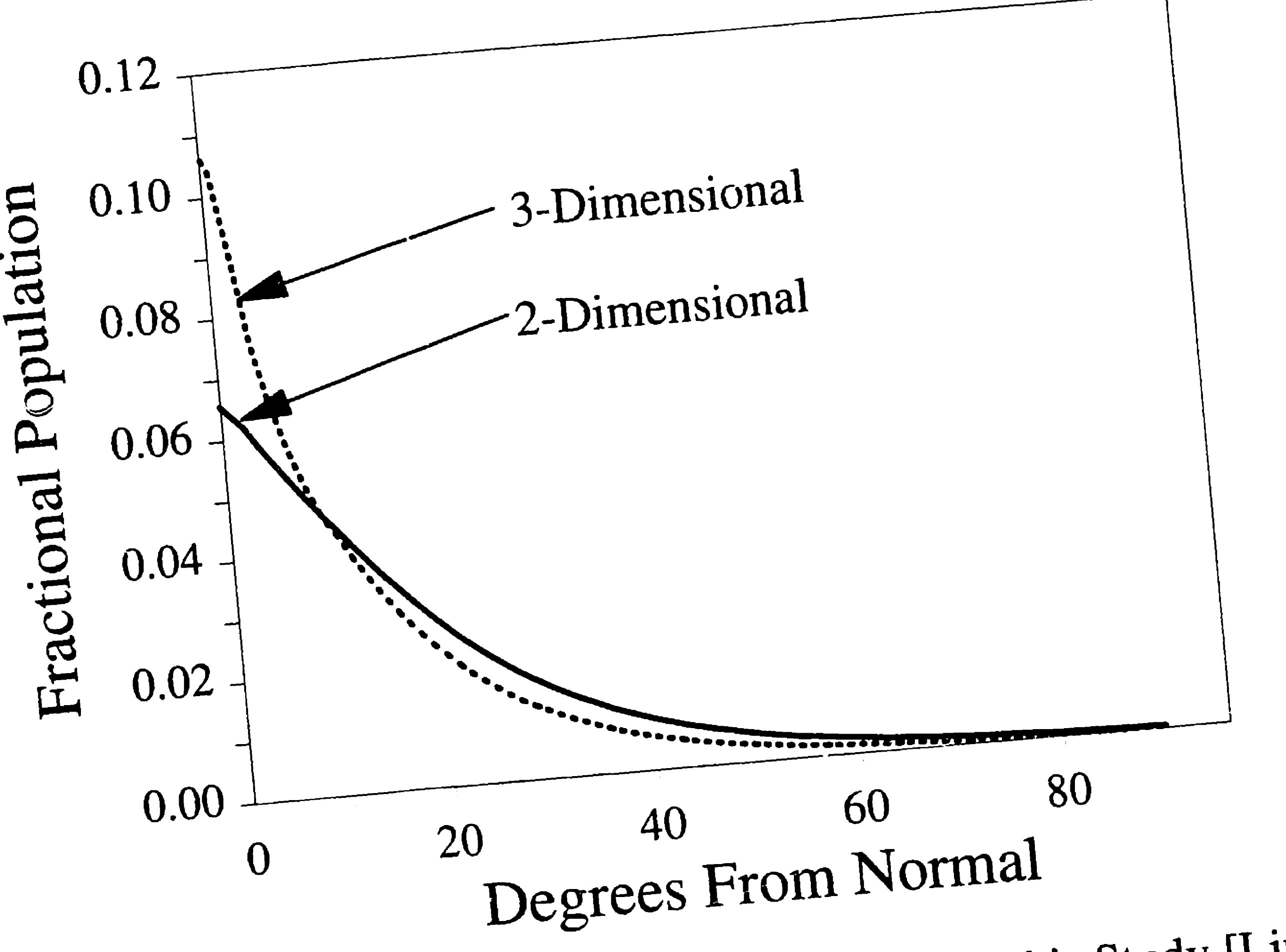


Figure 4-6: Ion Angular Distributions: The 3-D IAD used in this Study [Liu *et al.*, 1990] and the Equivalent Weighted 2-D IAD. The 2-D Distribution was Computed by Integrating Over the Azimuthal Angle. When Used in the Equivalent 2-D Model, It Computes the Correct Scattering Flux For the Specular Scattering Case.

The probability of reflection is a function of the ion energy, ion incidence angle, and the composition of the surface. In this work, we assume that the primary effect is the angle of incidence with respect to the microscopic surface feature. Ions arriving at grazing incidence should not transfer a significant fraction of their energy to the surface and are more likely to be reflected in a near specular fashion. As the incident angle approaches normal incidence, the ions should transfer a greater fraction of their energy to the surface, and may induce reactions and sputtering, or may be imbedded in the surface. Therefore, the reflection probability is expected to be close to unity at grazing incidence and decrease at higher angles, approaching zero at some incidence angle. Kelly and Auciello [1980] have formalized this by proposing the inequality

$$\hat{\theta} < \theta_R < \theta_c < \frac{\pi}{2} , \quad \textbf{(4-5)}$$

where $\hat{\theta}$ is the angle of the maximum sputtering yield, θ_R is the angle above which reflection is total, and θ_c is the angle at which the sputtering yield reaches zero. This leads to a general reflection probability function as shown in Figure 4-7 (following Wilson *et al.* [1984]).

The parameters $\hat{\theta}$, θ_R, and θ_c remain to be determined. In the Cl^+-Si system, physical sputtering is overshadowed by chemical reaction between the ion and the surface; this makes it difficult to determine $\hat{\theta}$ and θ_c. This reaction has a different angular dependence, [Mayer *et al.*, 1985],[van Zwol *et al.*, 1987] with a peak yield at normal incidence. The physical sputtering behavior of Cl^+ should be similar to that of

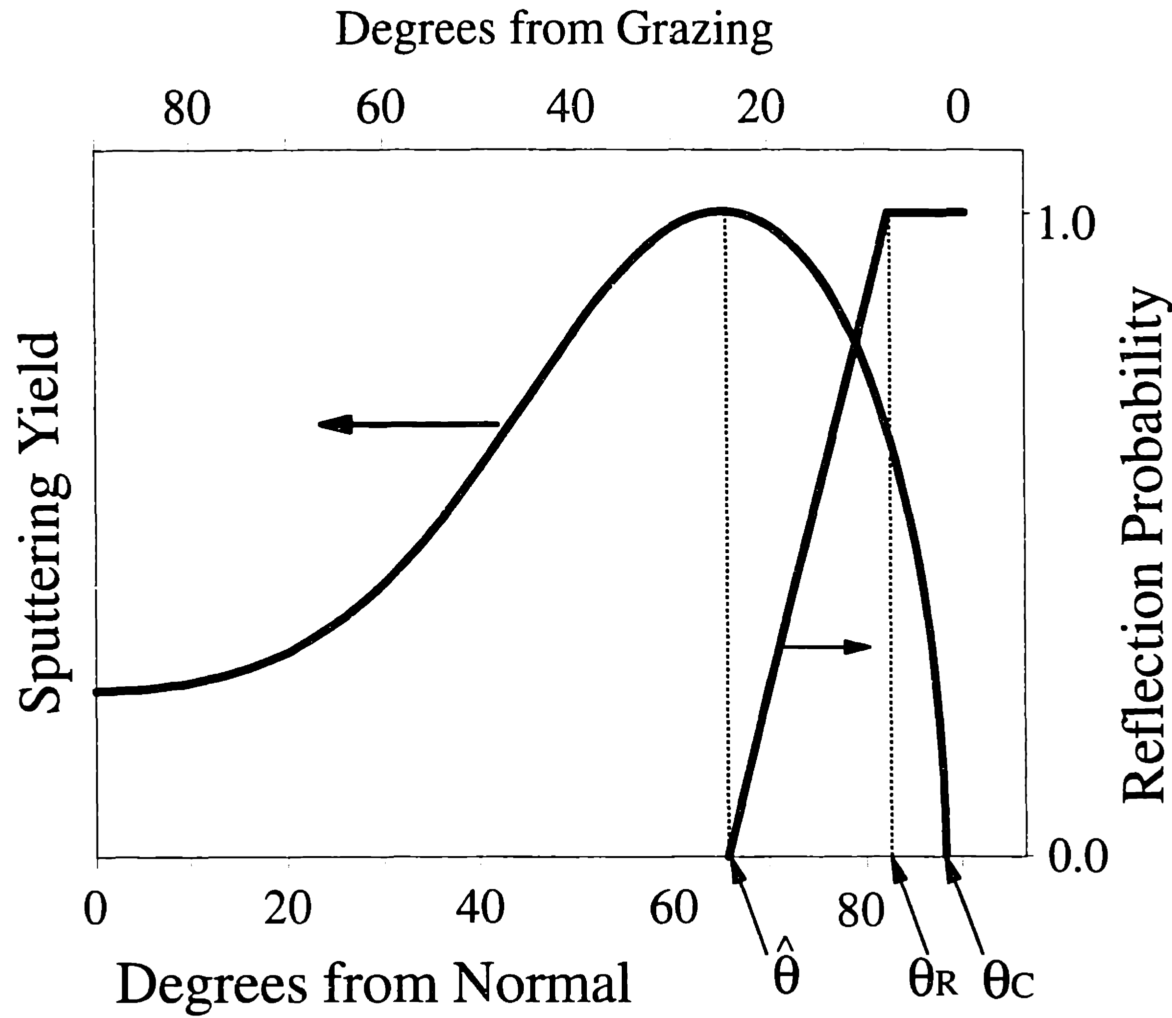


Figure 4-7: Physical Sputtering Yield and Assumed Reflection Probability as a Function of Ion Impingement Angle. Note: The Drop Off in the Sputtering Yield at Grazing Angles is Associated with Ion Reflection.

Ar^+, which has similar atomic number and mass but no obscuring chemical reactions. The angular dependence of the Ar^+-Si sputtering yield has been reported extensively, with typical $\hat{\theta}$ and θ_c values of 60-65° and 90°, respectively. [Ducommun *et al.*, 1975],[Smith *et al.*, 1987]. Hou and Robinson [1978] have calculated the reflection probabilities for several 3 keV ions incident on Copper planes, obtaining θ_R in the range of 75-85°. Moreno-Marin *et al.*, [1986] favored a linear approximation to the reflection probability (i.e. $\theta_R = 90°$).

In this work, θ_R was varied in the range 70-90°, while $\hat{\theta}$ and θ_c were fixed at 60° and 90°, respectively. The different probability functions resulted in changes in the magnitude of the reflected ion flux, but did not alter the qualitative trends in the ion flux along the feature bottom. All of the simulations presented in this paper use the intermediate value of θ_R, 80°. The identical function was also assumed for the reflection of ions from the photoresist, because the physical sputtering characteristics for silicon and photoresist are similar. [Rangelow, 1983].

The etching simulations were begun at polysilicon etching endpoint. As was previously mentioned, microtrenches did not form by the time endpoint was reached. They only formed during the more selective overetch step. The photoresist and polysilicon sidewall profiles did not change during the overetch step, probably due to the presence of a siliconoxychloride passivation layer. However the height of the photoresist changed during the overetch step as the resist top was eroded. Feature dimensions and angles were determined by measuring profiles obtained using a SEM.

4.4 Model Results

The computed ion flux (or etching rate) profile across the bottom of an etching feature (resist angle 84°, resist thickness .5μm, polysilicon angle 75°) is shown in Figure 4-8a. Distinct maxima occur in the ion flux about 560 Å from each sidewall. The maxima are not seen in any of the individual components Φ_D, Φ_P, or Φ_R, but are a result of the summation of all the fluxes. Therefore, the occurrence of microtrenching cannot be attributed to simple scattering from a single surface, but is a result of opposing trends in the fluxes from several surfaces. The scattered flux from each film is composed of overlapping fluxes reflected from each sidewall as shown in Figure 4-9, which is for a symmetric profile. Contributions from the two sidewalls would be different for an asymmetric profile.

In Figure 4-8a the terms used to characterize the microtrenching behavior are illustrated, i.e. the maximum and minimum fluxes/etching rates and the offset of the maximum ion flux from the sidewall. The maximum non-uniformity ratio is defined as

$$R_{max} \equiv \frac{ER_{max}}{ER_{min}} . \tag{4-6}$$

This parameter describes the likelihood of punch-through, and should be minimized for process reliability. The offset, defined as the distance separating the maximum etching rate from the sidewall, does not have specific process implications, but is useful in validation of the importance of ion scattering in the formation of the trenches.

Selected results of a parameter study with a nominal linewidth of 0.5 μm and polysilicon thickness fixed at 3200 Å are shown in Figures 4-10 and 4-11. Several trends

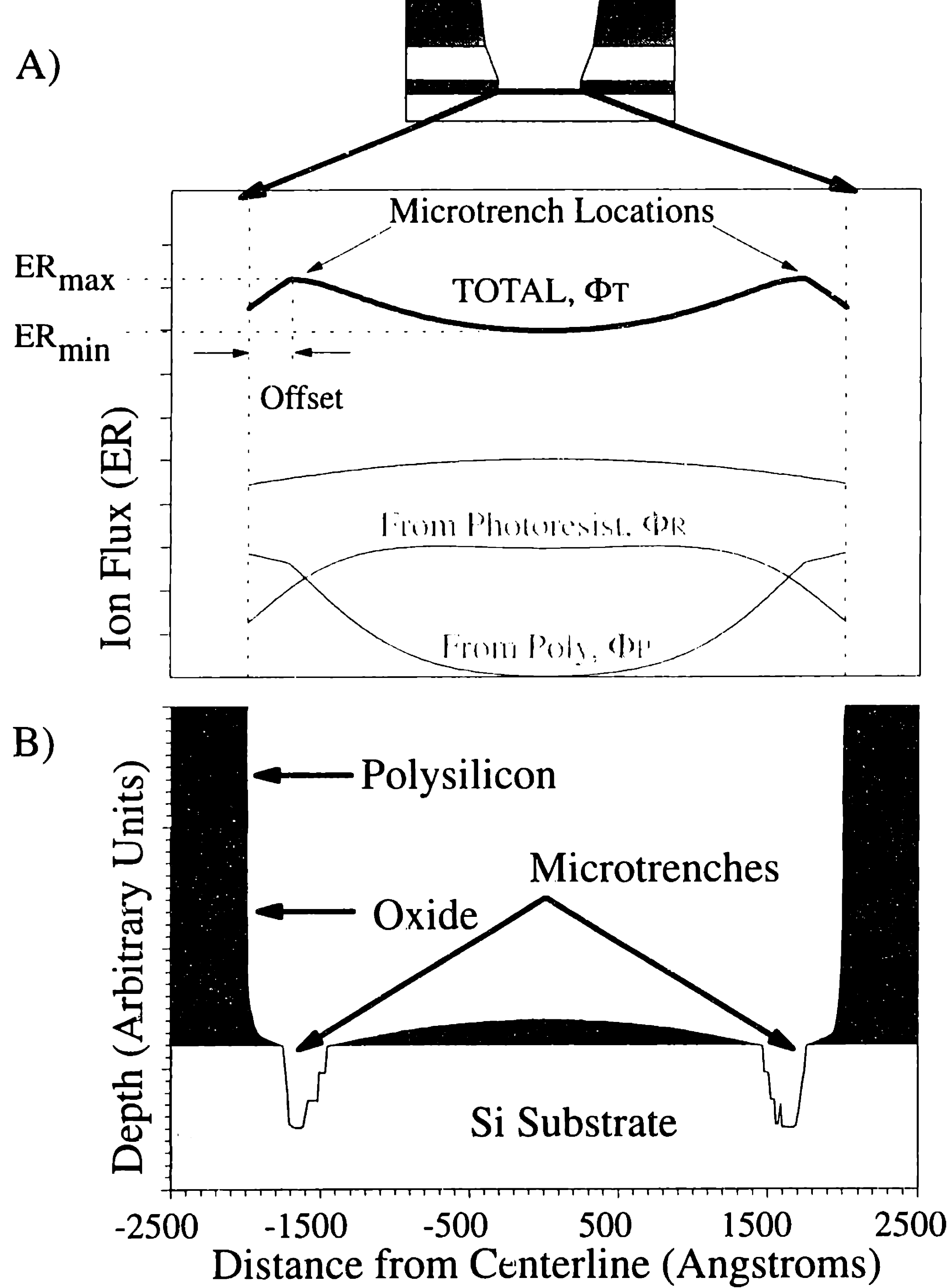


Figure 4-8 : Predicted Ion Flux Profile and Surface Topography For Structure Showin in Figure 1. (A) Initial Component and Total Ion Fluxes Along Feature Bottom. (B) Surface Topography After Etching. Note: Microtrench Locations Correspond to Maxima in Ion Flux.

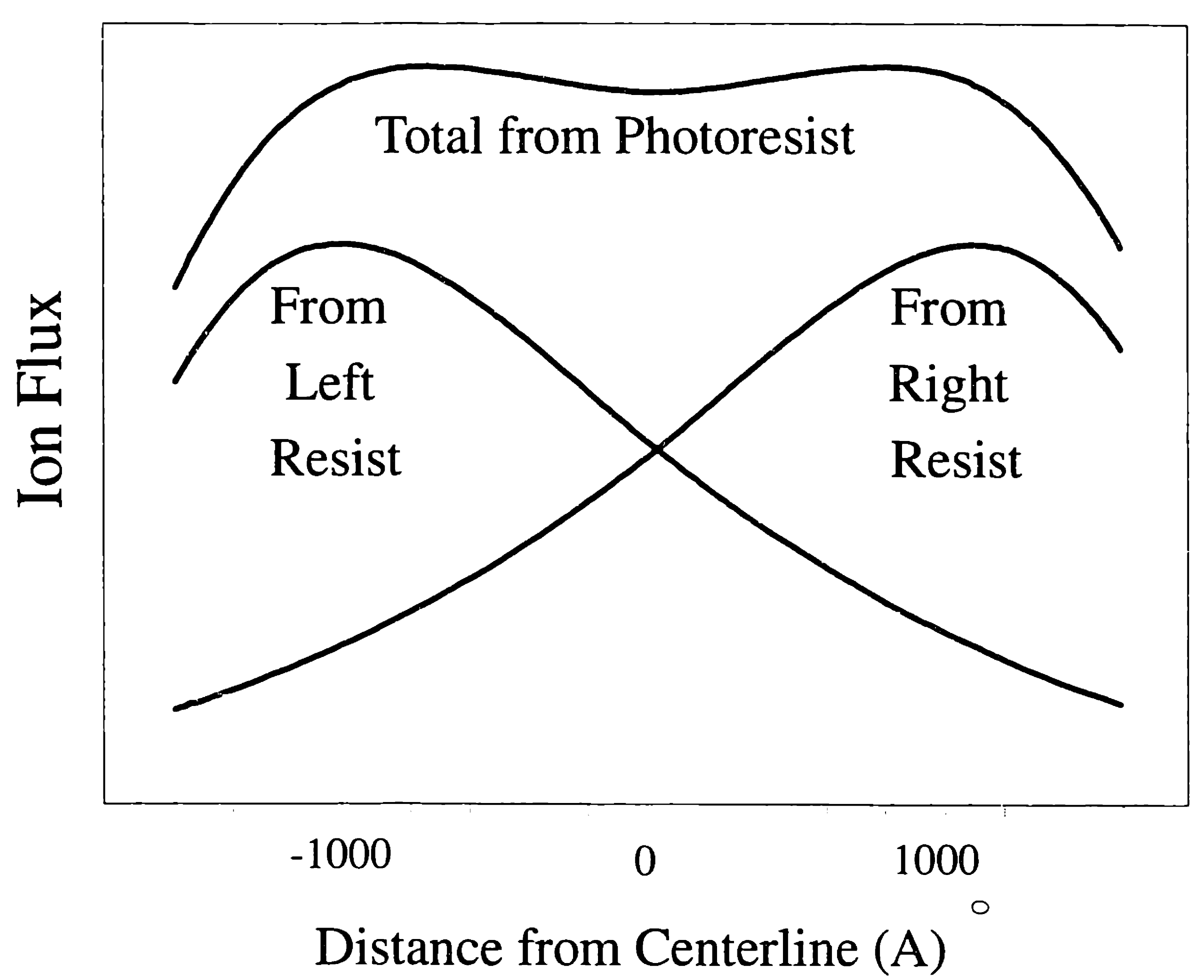


Figure 4-9: Details of Φ_R for Feature Modeled in Figure 4-8. Total Φ_R is the Sum of Ion Fluxes Reflected from Left and Right Sidewall Photoresist.

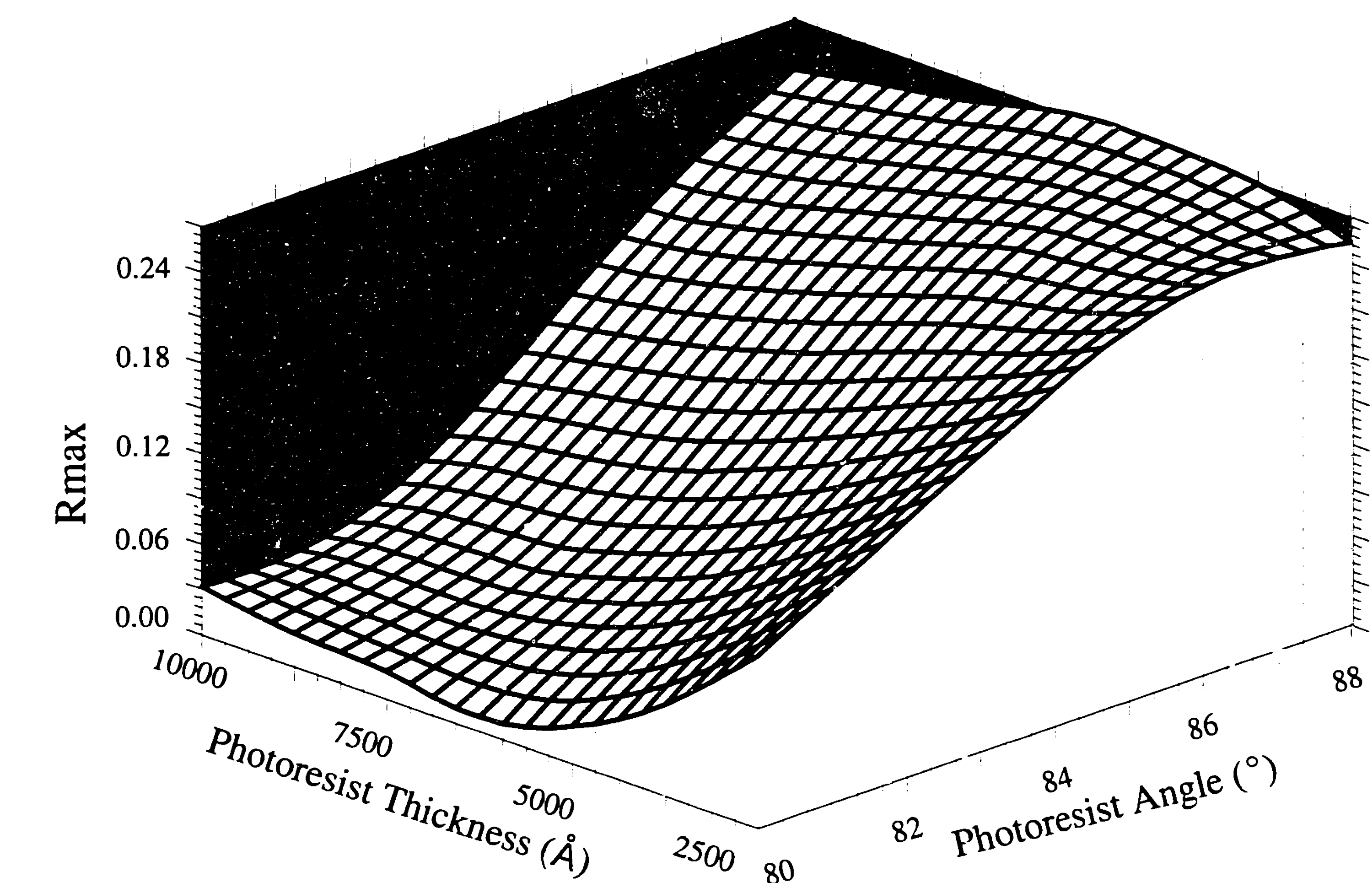


Figure 4-10: Response Surface for Maximum Nonuniformity Ratio, R_{max}, with Varying Photoresist Angle and Thickness. Linewidth, Polysilicon Angle, and Thickness Were Held Constant at 0.5 µm, 80°, and 3000Å, Respectively.

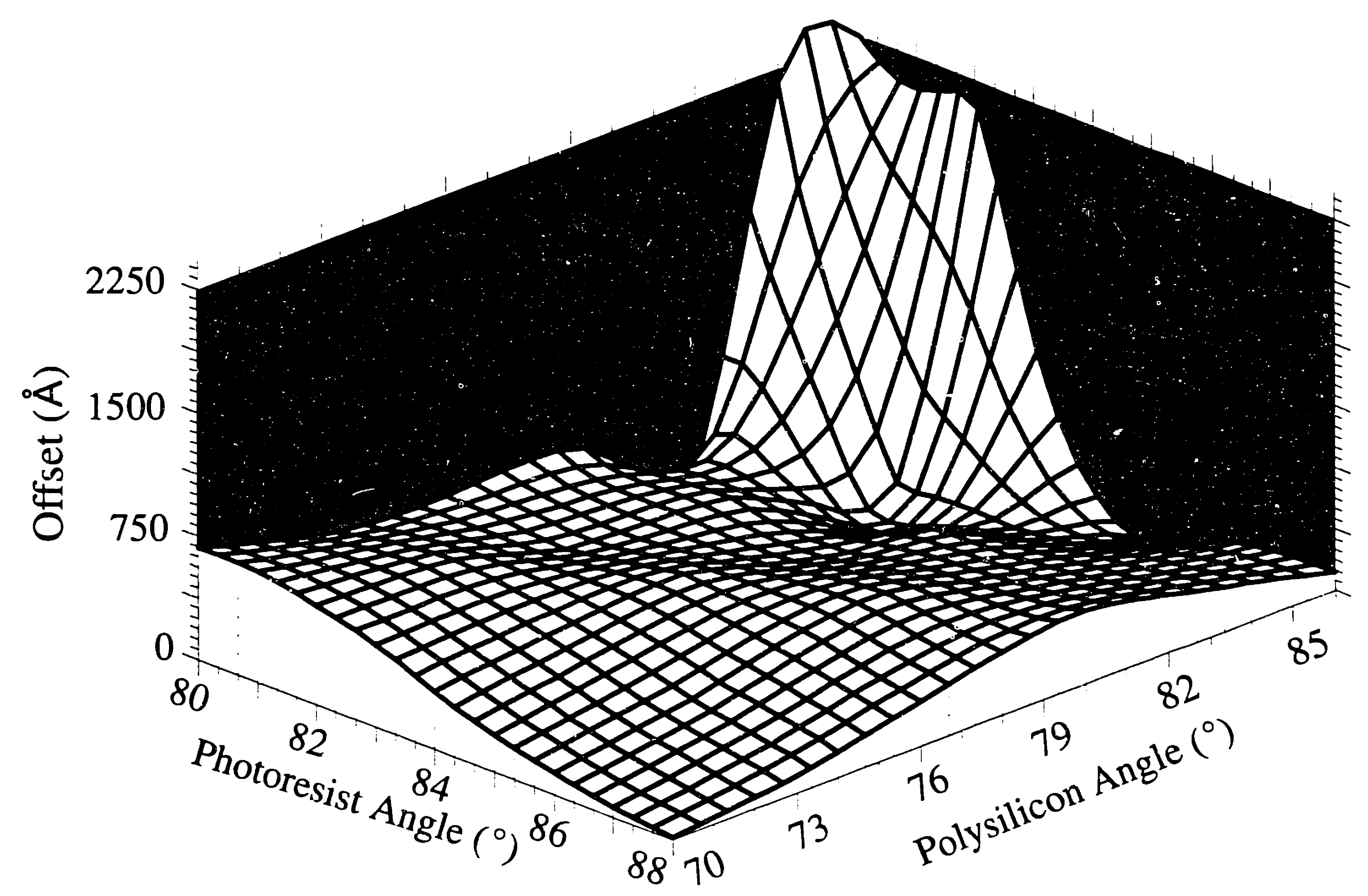


Figure 4-11: Response Surface for Microtrench Offset as a Function of Photoresist and Polysilicon Angles. Linewidth and Film Thickness Were Held Constant at 0.5 μm and 3200 Å, Respectively.

are evident; first, the degree of non-uniformity (R_{max}) decreases with decreasing photoresist angle. This is apparently due to distribution of the reflected flux over a wider portion of the bottom. Second, the variation of R_{max} with photoresist thickness is dependent upon the photoresist angle. The reflected particles distribute over a range which varies with resist thickness as

$$\Delta x = T_R \left[\frac{1}{\tan(\theta_R + \theta - \frac{\pi}{2})} - \frac{1}{\tan(\theta_R)} \right] + T_P \left[\frac{1}{\tan(\theta_R + \theta - \frac{\pi}{2})} - \frac{1}{\tan(\theta_P)} \right], \quad \text{(4-7)}$$

where Δx is the range along the bottom, T_R is the photoresist thickness, θ_R is the photoresist angle, T_P is the polysilicon thickness, θ_P is the polysilicon slope and $\hat{\theta}$ is the maximum incidence angle for reflection (60° from Normal).

At high angles (nearly vertical walls) and with thin resist, the non-uniformity increases with thickness. This is because more ions strike the taller sidewall, yet they all reflect to a small region of the bottom. This trend is reversed at lower resist angles.

At higher resist thicknesses, R_{max} is determined by the overlap of fluxes from the two sidewalls. For a given combination of sidewall angles and line spacing S, there will be critical resist thickness above which the reflected flux spans the entire feature bottom. This thickness is defined by

$$T_R = \frac{\tan(\theta_R + \theta - \frac{\pi}{2})\tan(\theta_R)}{\tan(\theta_R) - \tan(\theta_R + \theta - \frac{\pi}{2})} \left[S - T_P \left(\frac{1}{\tan(\theta_P)} + \frac{1}{\tan(\theta_R + \hat{heta} - \frac{\pi}{2})} \right) \right] \quad \text{(4-8)}$$

For the conditions shown in Figure 4-10, the critical thicknesses are between about 6700 and 7500 Å. As seen in the figure, R_{max} varies with increasing photoresist

thickness up to this point, and is relatively constant thereafter.

The final noteworthy trend is the location of the microtrench, as described by the offset parameter in Figure 4-11. As the resist sidewalls approach vertical, the microtrench moves closer to the sidewall. This provides a possible explanation for the discrepancy between trench locations for the present work and previous reports; i.e. the resist and polysilicon sidewalls in this experiment are less vertical.

4.5 Comparison to Experiment

Comparison of model and experimental results required extraction of film angles and thicknesses from SEM's. The accuracy of these measurements was limited, but reasonable estimates were obtained. The trench shown in Figure 4-1 had a photoresist angle and thickness of 86° and 1 μm, respectively, and a polysilicon angle of 81°. Figure 4-8 shows the model output for such a structure. Figure 4-8a contains the calculated etching rate profile across the trench bottom. Figure 4-8b contains the predicted surface profile after 1.9 minutes of etching, in good agreement with the actual profile in Figure 4-1. The location and width of the microtrench are correctly predicted. Detailed prediction of the microtrench shape after punch-through is limited by numerical resolution used in these simulations.

Some of the experimental samples displayed asymmetry, as illustrated in Figure 4-12a. Careful measurement of the resist and polysilicon profiles showed that they were asymmetric; the resist angle on the right side was 84° (vs 86° on the left), and the polysilicon angle was 77° (vs. 83°). Figures 4-12b and 4-12c show the model outputs for

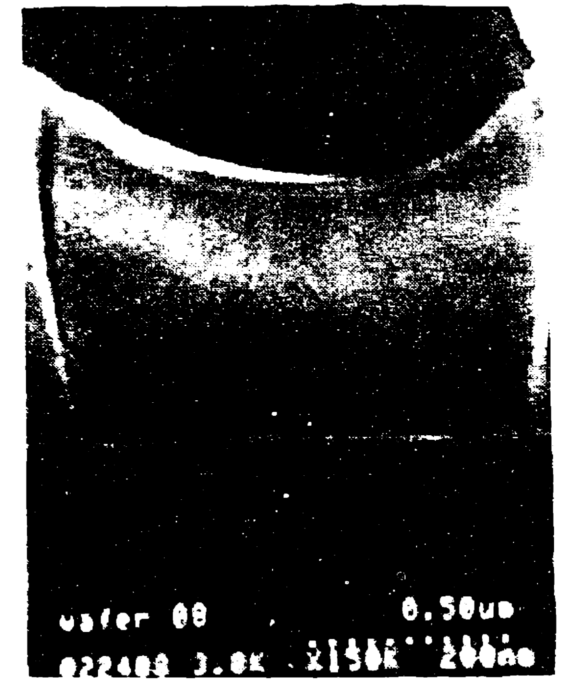


Figure 4-12A : Scanning Electron Micrograph of a Feature Cross Section Showing Asymmetric Microtrenching.

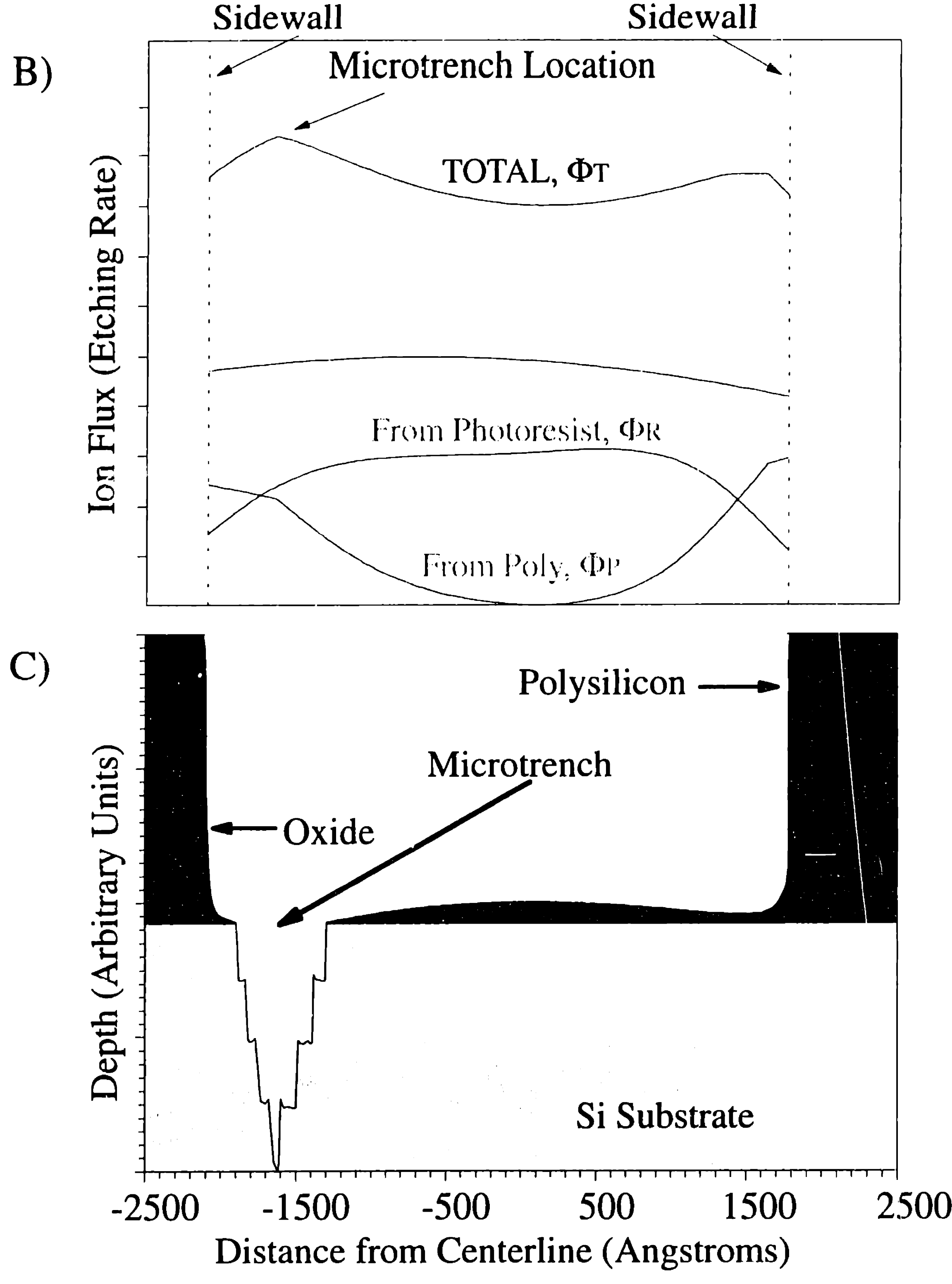


Figure 4-12 B,C: Modeling of Asymmetric Microtrenches. (B) Predicted Initial Ion Fluxes Along Feature Bottom. (C) Surface Topography After Etching. Note: Microtrench Locations Correspond to the Maximum in the Flux Profile.

this structure, again reproducing the major experimental observations. Figure 4-12a shows a large microtrench on the left-hand side of the trench, and a much smaller one on the right-hand side. The total ion flux profile in Figure 4-12b shows similar behavior, with a global maximum on the left hand side and a smaller, local maximum on the right-hand side.

The experimental study also considered the effect of decreasing space width, with spaces as small as 0.35 μm. As was shown previously in Figure 4-9, the fluxes of ions reflecting from the two sidewalls overlap on the trench bottom. As the space width decreases, the degree of overlap increases. The two maxima in the flux profile come closer together, finally merging into a single maximum at the trench centerline. This is illustrated in Figure 4-13b, in which the total flux profiles for trench widths ranging from 0.9 μm down to 0.3 μm are plotted. Figure 4-13a shows the SEM for a 0.35 μm space.

4.6 Conclusions

An unusual microtrenching phenomenon was observed in which the trench was offset from the sidewall. A simple numerical model for the specular reflection of ions from feature sidewalls provided nearly quantitative agreement with the experimental results. This model also reproduced the qualitative characteristics of previously reported microtrenches, leading to the conclusion that ion reflection is a probable cause of microtrenching under some conditions. Further investigations of microtrenching during plasma etching will be presented in the future in the thesis of John Arnold [Arnold, 1994].

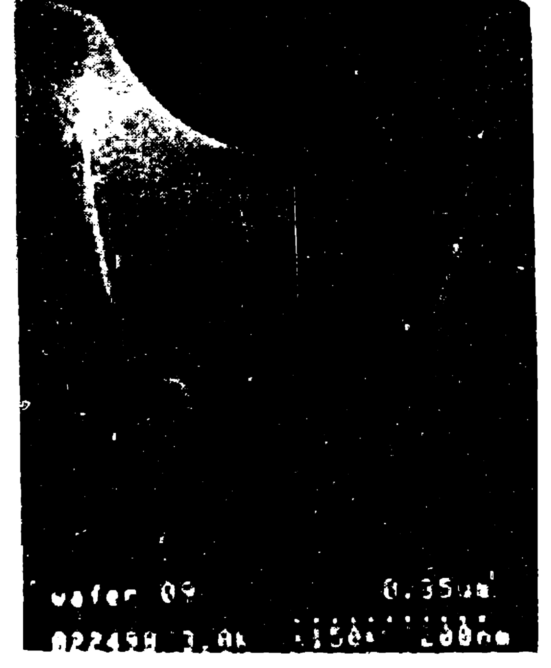


Figure 4-13A : Effect of Line Spacing on Microtrenching. (A) Scanning Electron Micrograph of a 0.35 µm Feature, Showing a Single Microtrench at the Centerline.

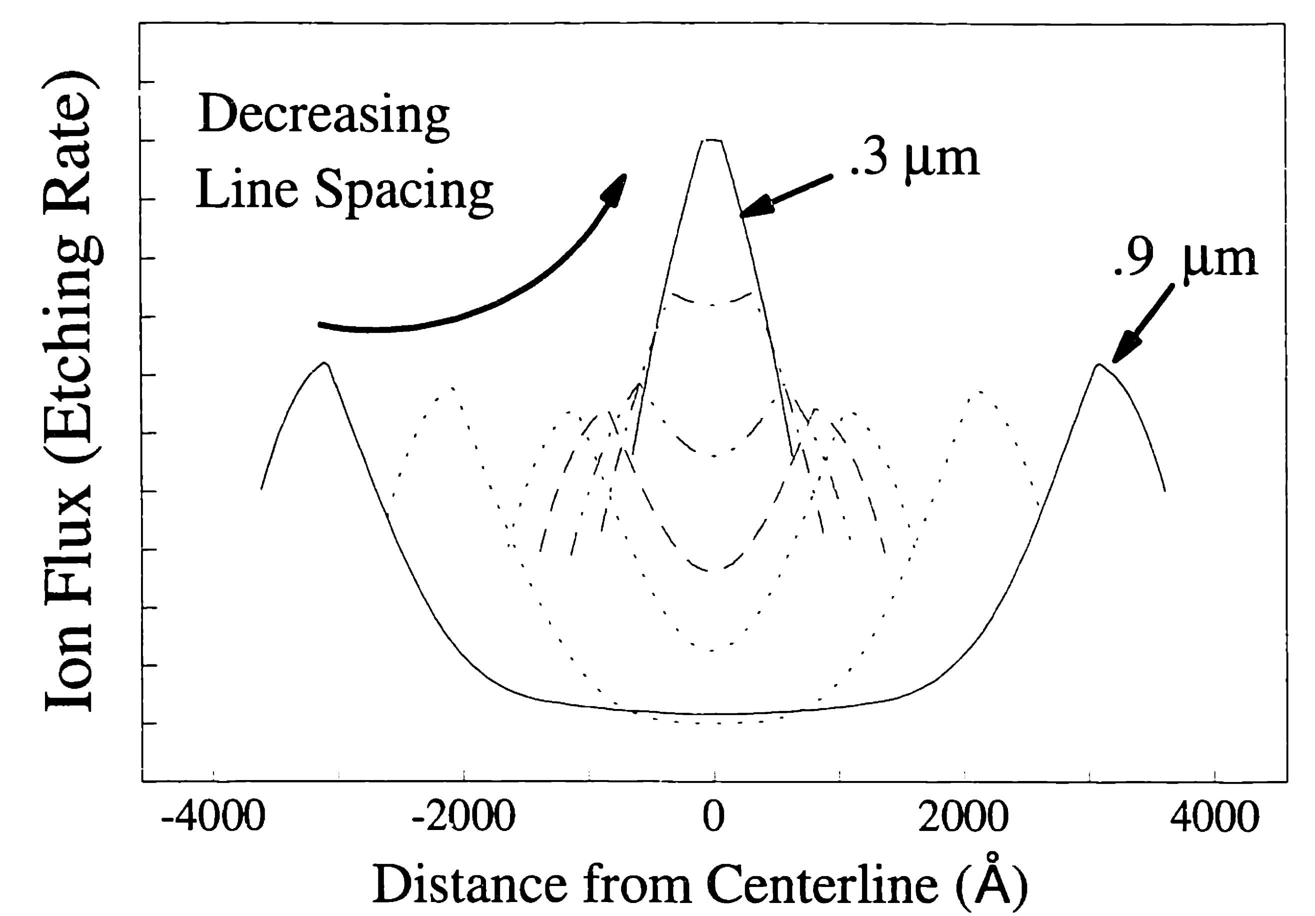


Figure 4-13B : Effect of Line Spacing on Microtrenching. (B) predicted Initial ion Flux Profiles Where Film Angles and Thicknesses Are Held Constant. As Space Width Decreases, Reflected Fluxes from the Two Sidewalls Overlap and Merge, Forming a Single Maximum Near the Centerline. Shown are Line Spacings of 0.30, 0.35, 0.40, 0.45, 0.50, 0.70, and 0.90 μm.

Chapter 5

Full Wafer Interferometry

A new technique has been developed to measure etch rate uniformity *in situ* using a CCD camera which views the wafer during plasma processing. The technique records the temporal modulation of plasma emission or laser illumination reflected from the wafer; this modulation is caused by interferometry as thin films are etched or deposited. The measured etching rates compared very well with those determined by Helium-Neon laser interferometry. This technique is capable of measuring etch rates across 100-mm or larger wafers. It can resolve etch rate variations across a wafer or within a die. The CCD measurement technique is a valuable tool for process development and has potential use as a real-time diagnostic for process control.

5.1 Introduction

Uniformity of etching has always been an important issue in plasma etching. Wafer-to-wafer uniformity within a batch is no longer a major concern because multi-wafer batch reactors are being replaced with single-wafer etchers; the dominant concern today is etching process uniformity across a single wafer, especially as silicon wafer diameters are increasing [Coburn, 1986]. State-of-the-art wafer processing is done on 200-mm diameter wafers. The next generation of wafers will probably be 250 and/or 300-mm diameter. Presently, uniformity is determined *ex situ* by optically measuring

film thickness (for example, with a Nanospec, or Prometrix) before and after a partial etch of known duration [Nagy, 1984],[Kao and Stenger, 1990]. This process is not suitable as a diagnostic for real-time process control.

A number of techniques exist to measure etching rate and/or film thickness *in situ*; among these are laser interferometry (LI) [Busta *et al.*, 1981],[Marcoux and Foo, 1981],[Sternheim *et al.*, 1983], optical emission interferometry (OEI) [Heinrich *et al.*, 1989],[Angell and Oehrlein, 1991], and ellipsometry [Henck, 1992],[Henck et al., 1993]. Laser interferometry and optical emission interferometry are both governed by interference of light reflected from a thin film, but they use different light sources. Laser interferometry uses a laser beam (typically a 632.8 nm Helium-Neon (HeNe)) while optical emission interferometry uses plasma optical emission as the light source. Ellipsometry measures the change in polarization of light upon its reflecting from a surface.

5.1.1 Physics of Interferometry

Figure 5-1 depicts a thin film of thickness d and refractive index n_2 on a substrate with index n_3. Light rays at a wavelength of λ travel through a medium with index n_1 and strike the thin film at an angle θ_1 relative to the surface normal and are both reflected (ray a) and refracted inside the film to an angle θ_2. Following the development of Hecht & Zajac [1979], the optical path difference, Δl_{opt}, between light reflected from the film (a) and light that has traveled through the film and been reflected from the lower interface (b) can be written as

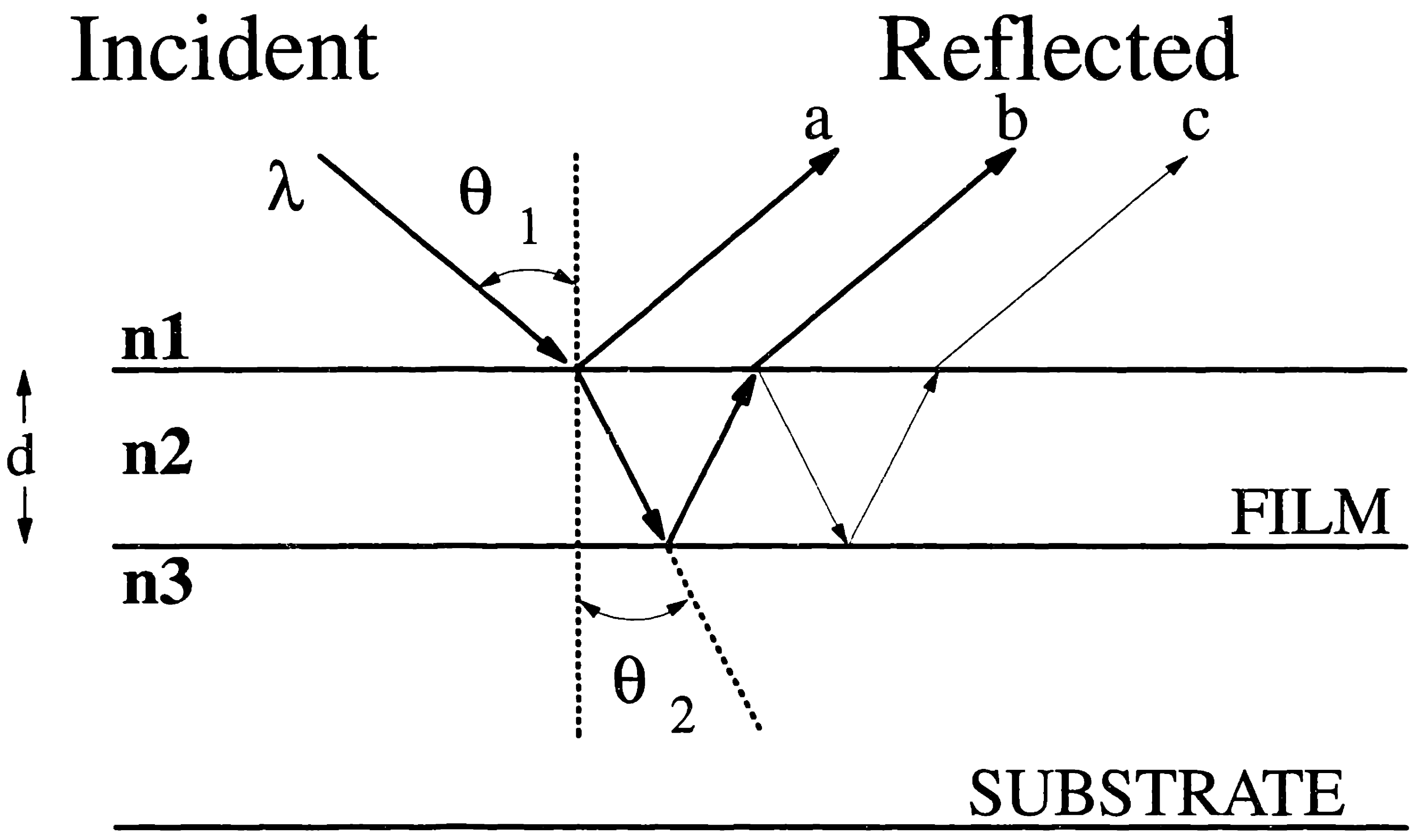


Figure 5-1: Film structure and variables used in interferometry calculations.

$$\Delta l_{opt} = \frac{2n_2 d}{\cos(\theta_2)} - 2n_1 d \tan(\theta_2) \sin(\theta_1) . \qquad (5\text{-}1)$$

Light that undergoes multiple reflections (c) within the film is usually sufficiently low in intensity that it can be neglected. Using Snell's Law, Equation (1) can be written in terms of θ_2 alone, and simplified to yield:

$$\Delta l_{opt} = 2n_2 d \cos(\theta_2) . \qquad (5\text{-}2)$$

A maximum in the interference between the two reflected light rays (a and b) will occur when the optical path difference is an integral multiple of the wavelength of the light. Taking n_1 to be 1, Δd_{film}, the change in film thickness between adjacent maxima (or adjacent minima), is given by

$$\Delta d_{film} = \frac{\lambda}{2n_2 \cos(\theta_2)} . \qquad (5\text{-}3)$$

For normal incidence, this reduces to simply

$$\Delta d_{film} = \frac{\lambda}{2n_2} . \qquad (5\text{-}4)$$

The term $\cos(\theta_2)$ is a correction for non-normal incidence, which can be rewritten in terms of measurable parameters as

$$\cos(\theta_2) = \sqrt{\left(1 - \left[\frac{\sin(\theta_1)}{n_2}\right]^2\right)} . \qquad (5\text{-}5)$$

The refractive index, n_r, is real for a transparent film and is complex ($n = n_r + ki$) for an

absorbing film, where k is the absorption coefficient, and $i^2 = -1$. The indices of refraction for single crystal silicon and glassy silicon dioxide are listed in Table 5-1 [Edwards, 1985],[Phillip, 1985].

Table 5-1

Complex Indices of Refraction for Polysilicon and SiO2

Material	Wavelength (nm)	n_r	k
Si	477.3	4.439	0.094
Si	632.8	3.882	0.019
Si	753.4	3.73	0.009
SiO_2	4.473	1.464	0.00
SiO_2	632.8	1.45705	0.00
SiO_2	753.4	1.45429	0.00

5.1.2 Interferometry Techniques

Two practical laser interferometry systems are shown in Figure 5-2. One is a laser interferometry system. A HeNe laser beam strikes the wafer surface. The reflected beam is directed through a bandpass filter to a photodiode, where the interferometry signal is recorded as a function of time. The bandpass filter prevents plasma emission from entering the photodiode while allowing the reflected laser beam to strike the photodiode. Similarly, an optical emission interferometry system is also shown. A lens collects

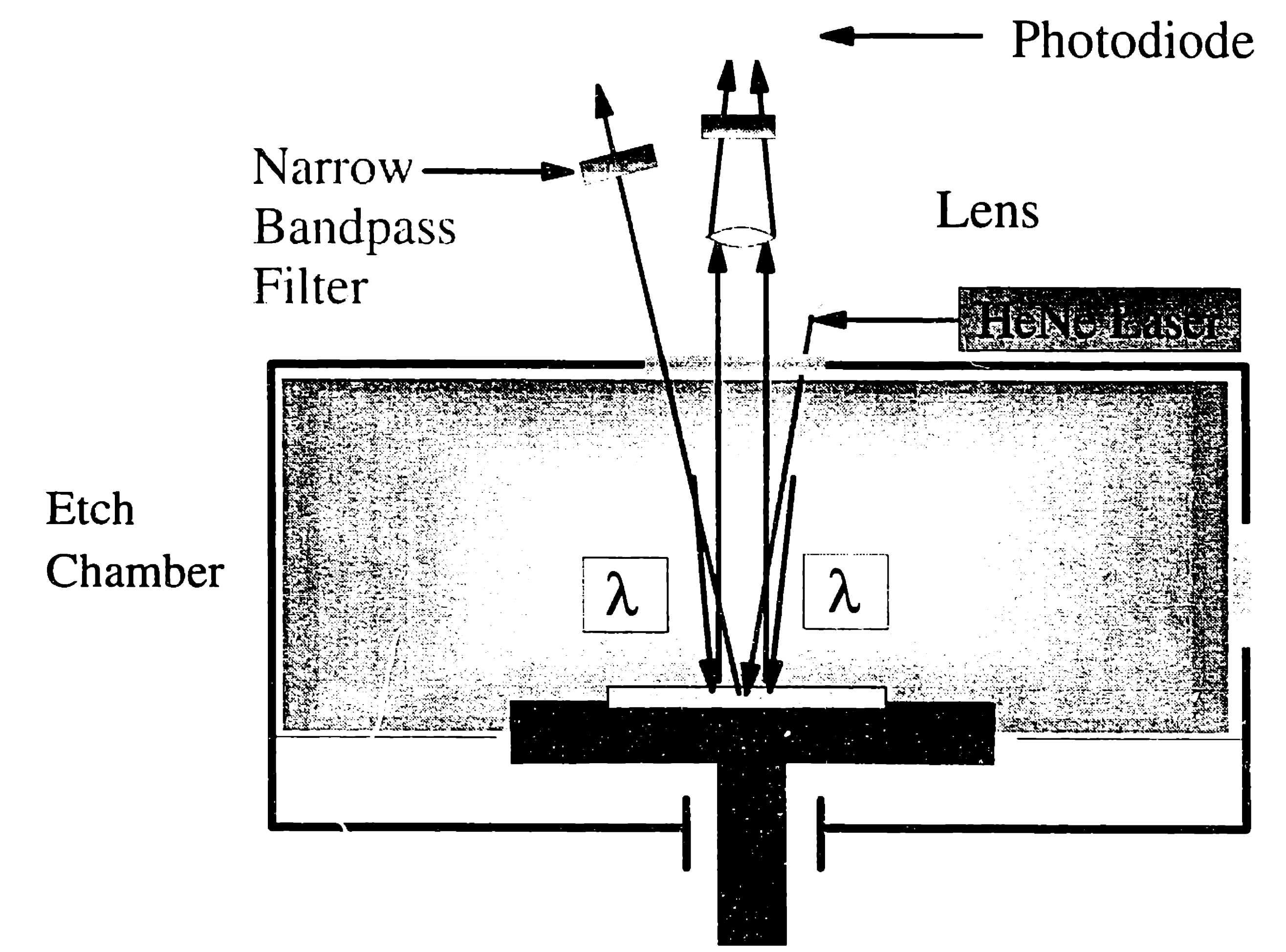


Figure 5-2: Systems for *in situ* Measurement of Etch Rate During Plasma Etching

plasma optical emission, which then passes through a bandpass filter and into a photodiode. The photodiode is typically located at the focal point of the lens. This produces an object distance of infinity since

$$\frac{1}{f} = \frac{1}{o} + \frac{1}{i} \tag{5-6}$$

where f is the lens focal length, o is the object distance and i is the image distance. As a consequence of the photodiode's placement, light is collected from over a small region on the wafer and the average of this light impinges on the photodiode. This arrangement does not allow for wafer imaging. In the OEI system, the bandpass filter defines the wavelength of light being used for interference and blocks light at unwanted wavelengths to prevent the plasma background from reaching the photodiode. The etching rate is calculated as

$$ER = \frac{\Delta d_{film}}{\delta_{tm}} \tag{5-7}$$

where δ_{tm} is the measured time between minima in the interferometry signal.

Both optical emission and laser interferometry have been used to monitor uniformity *in situ*. Davies *et al.* [1990] monitored photoresist thickness using a white light excitation source and a photodiode spectrometer array. Etching rate could have been determined at up to five points on a wafer; however, this was not done (the method was mentioned, but only single-point etching rate measurements were made). Economou *et al.* [1991] measured uniformity *in situ* at five points across a wafer by multichannel laser interferometry. However, their technique is not readily applicable to industrial plasma etching tools. It requires *a-priori* selection of sites for analysis and the splitting

and alignment of multiple beams. Also, the multichannel laser interferometry technique required an optical viewport the same diameter as the wafer being measured to obtain a full wafer view; viewports on industrial etchers are never this large. In this chapter, we describe the use of a CCD (charge couple device) camera to measure the optical emission interferometry across a wafer. The use of *in situ* ellipsometry to measure etching rates is fairly new; extension of the technique to measure uniformity is not obvious.

5.2 Experimental Method

A measurement technique has been developed in which etching rate uniformity is determined in real-time *in situ*. A CCD camera and an optical bandpass filter are used to image the wafer and monitor the interference of plasma optical emission light from a thin film on a wafer surface during plasma etching. Spatial and temporal variations in etching rates (and thus selectivity and uniformity) can be determined from the CCD images. The CCD technique allows for monitoring of the entire surface of a wafer; analysis points can be chosen both before and after the etch. Optical alignment is simple. Small optical access ports can be used to image large wafers, e.g. a two-inch window can be used to image an eight-inch wafer.

5.2.1 CCD SYSTEM

The experimental system is shown in Figure 5-3. An Electrim EDC-1000HR 8-bit CCD camera and acquisition board was used along with a 33 MHz 80386 IBM

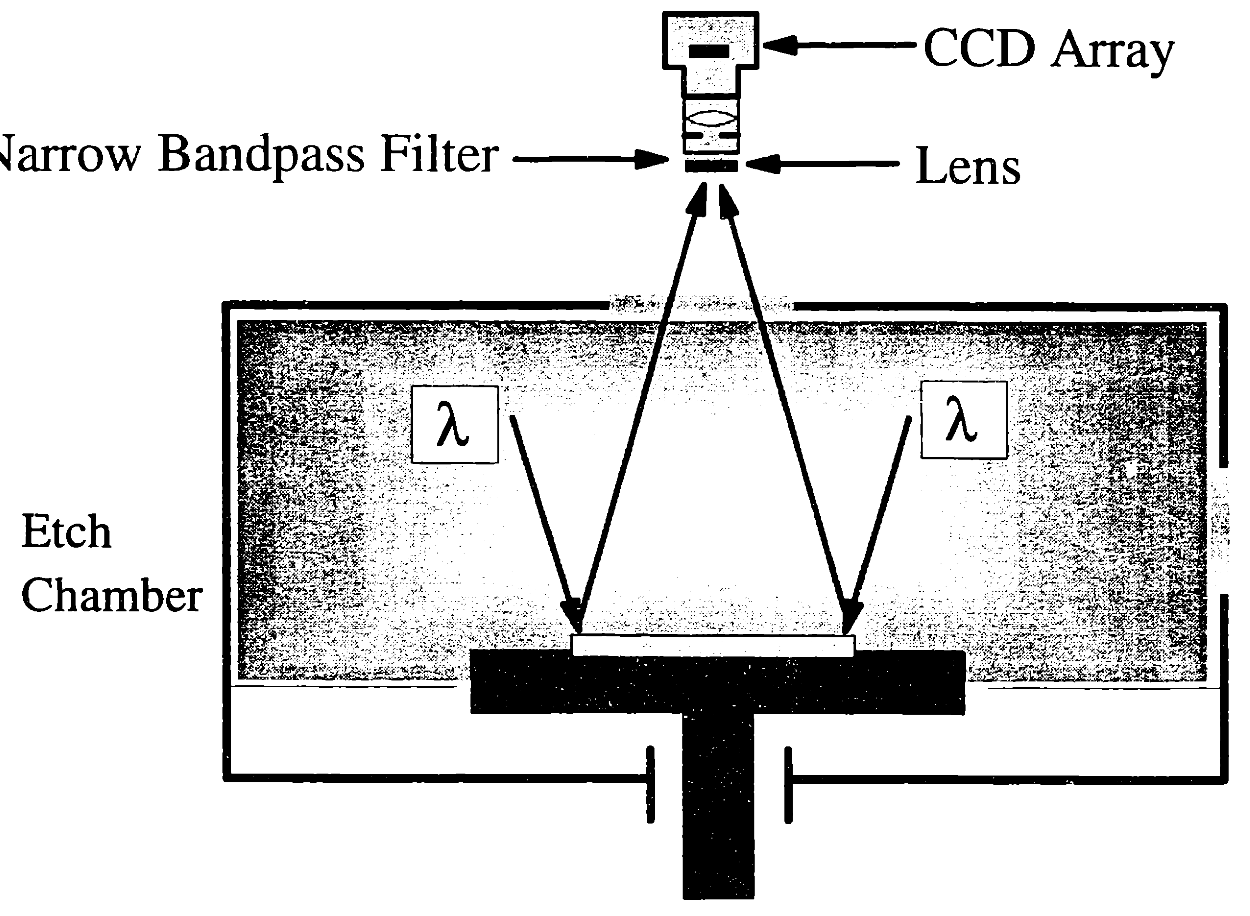


Figure 5-3 : CCD-based Full Wafer Interferometry System

compatible personal computer. A Texas Instruments TC241-22 CCD chip, which had a resolution of 756 x 244 pixels (756 x 488 in interlaced mode), was used in the camera. The CCD pixel dimensions were 11.5 x 27.0 μm and the chip was covered with a quartz window to allow for transmission of light in the UV region of the spectrum. A 16-mm focal length f/1.6 C-mount lens was used to completely view a 100 mm wafer through a 50-mm diameter optical viewport. Using this lens, a pixel on the CCD typically mapped to a surface area of approximately 180 x 420 μm. A Computar 50-mm f/1.3 lens was used for a detailed view of etching rates across a single 1 cm x 1 cm die. In this case, a pixel on the CCD typically mapped to a surface area of 27 x 63 μm. Pixel resolution can be improved by using a higher resolution CCD such as one with 1024 x 1024 or 2048 x 2048 pixels. A 4.8-mm f/1.8 lens can be used to observe a 200-mm wafer through a 50-mm viewport. Software was written to store multiple images from the EDC-1000HR in real time, and to analyze the data (see Appendix A.3).

5.2.2 CCD Interferometry

Plasma optical emission consists of a wide spectrum of light with wavelengths ranging from ultraviolet to near infrared. Interferometry can be done at various wavelengths. For OEI, shorter wavelengths are more desirable because the distance between maxima corresponds to a smaller film thickness, as is seen in Equation 4. A narrow optical bandpass filter was used to select the wavelength of light entering the CCD. The CCD had an acceptable quantum efficiency between 400 and 850 nm, with a

peak near 675 nm. Considering this and the location of strong emissions from the plasma, a bandpass filter (Andover Corporation, P/N 750FS40-25) with a center wavelength of 753.4 nm and a FWHM of 32.0 nm was selected. This filter transmits strong emissions from F, Cl, Br and Ar at 755.2, 754.7, 751.3 and 750.4 nm, respectively [Reader and Corliss, 1980]. The optical emission spectrum for the Cl_2/HBr etching chemistry employed is shown in Figure 5-4A for the wavelength range of 7000 to 8000 Å. Also plotted is the percent transmission of the 753.4 nm filter. Several of the strongest optical emission lines have been identified on this plot. The halogens fluorine, chlorine and bromine are important etchant gases; argon is used in many plasmas to increase ion bombardment and is frequently added to plasmas for use as an actinometer to determine relative concentrations. A near-UV filter (404.7 nm) was tried in order to observe smaller changes in film thickness. However, the plasma emission at this wavelength was not strong enough for interferometry to be observed with the CCD.

A more suitable optical emission interferometry filter in the blue-end of the spectrum was one with a peak transmission near 480 nm (LES, Somerville, MA). The filter transmission characteristics and the OE spectrum for the HBr/Cl_2 chemistry used (with the significant emission lines identified) are shown in Figure 5-4B for this filter. The intensity-weighted average wavelength is around 477.3 nm. Sample results from this filter are shown in Figure 5-7B and discussed later.

A bandpass filter at 632.8 nm was used to perform laser interferometry with the CCD camera as the detector. A 0.8-mm HeNe laser beam ($1/e^2$ diameter) was expanded to approximately 1 cm and used to illuminate a die. This technique showed significant improvements in signal-to-noise over OEI [Section 5.3.1]. However, this technique was

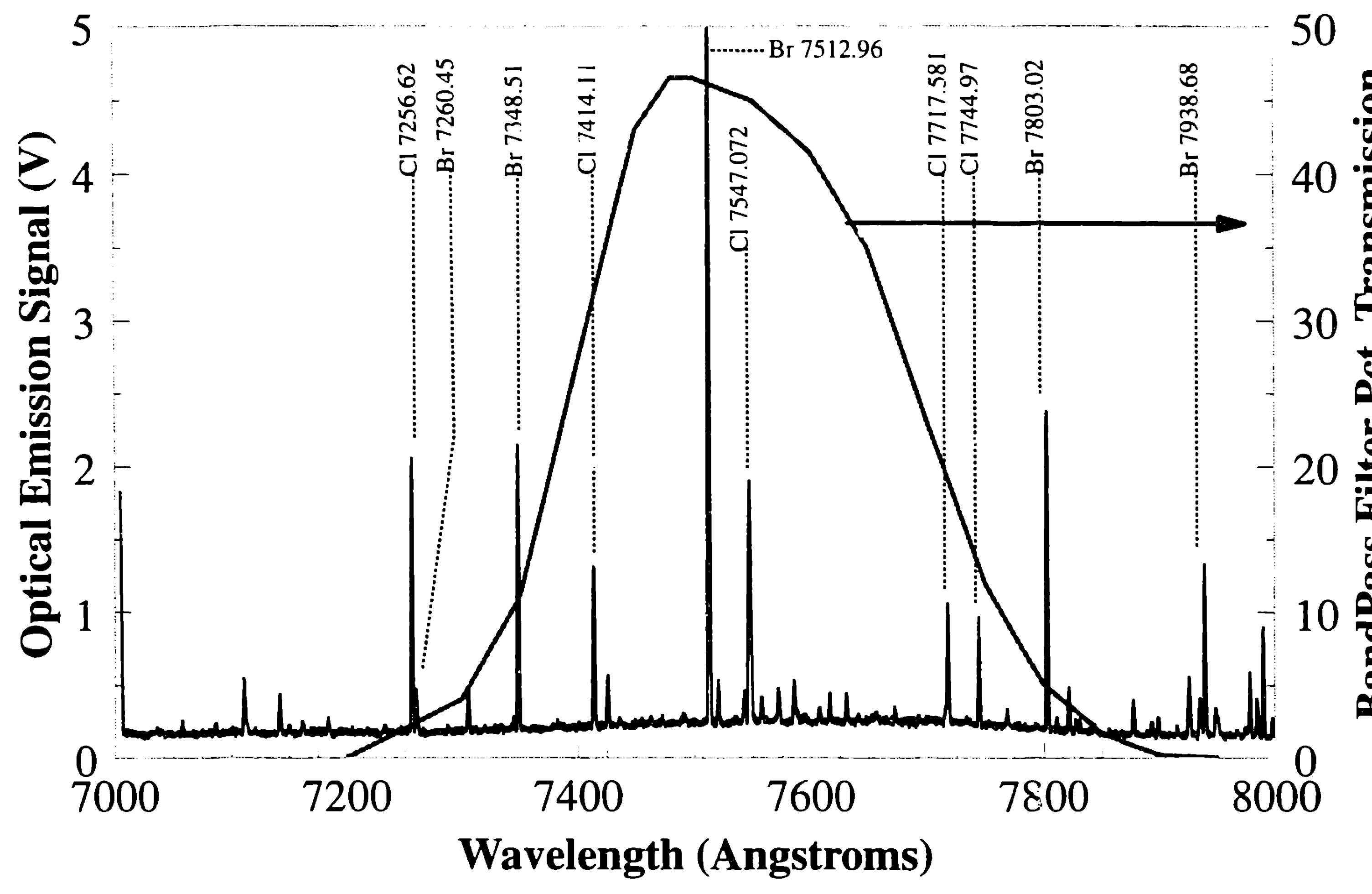


Figure 5-4A: Plasma Optical Emission Intensity for HBr/Cl_2 Plasma Etching. Also shown is the Optical Band Pass Filter Percent Transmission for the 753 nm Filter Used in FWI.

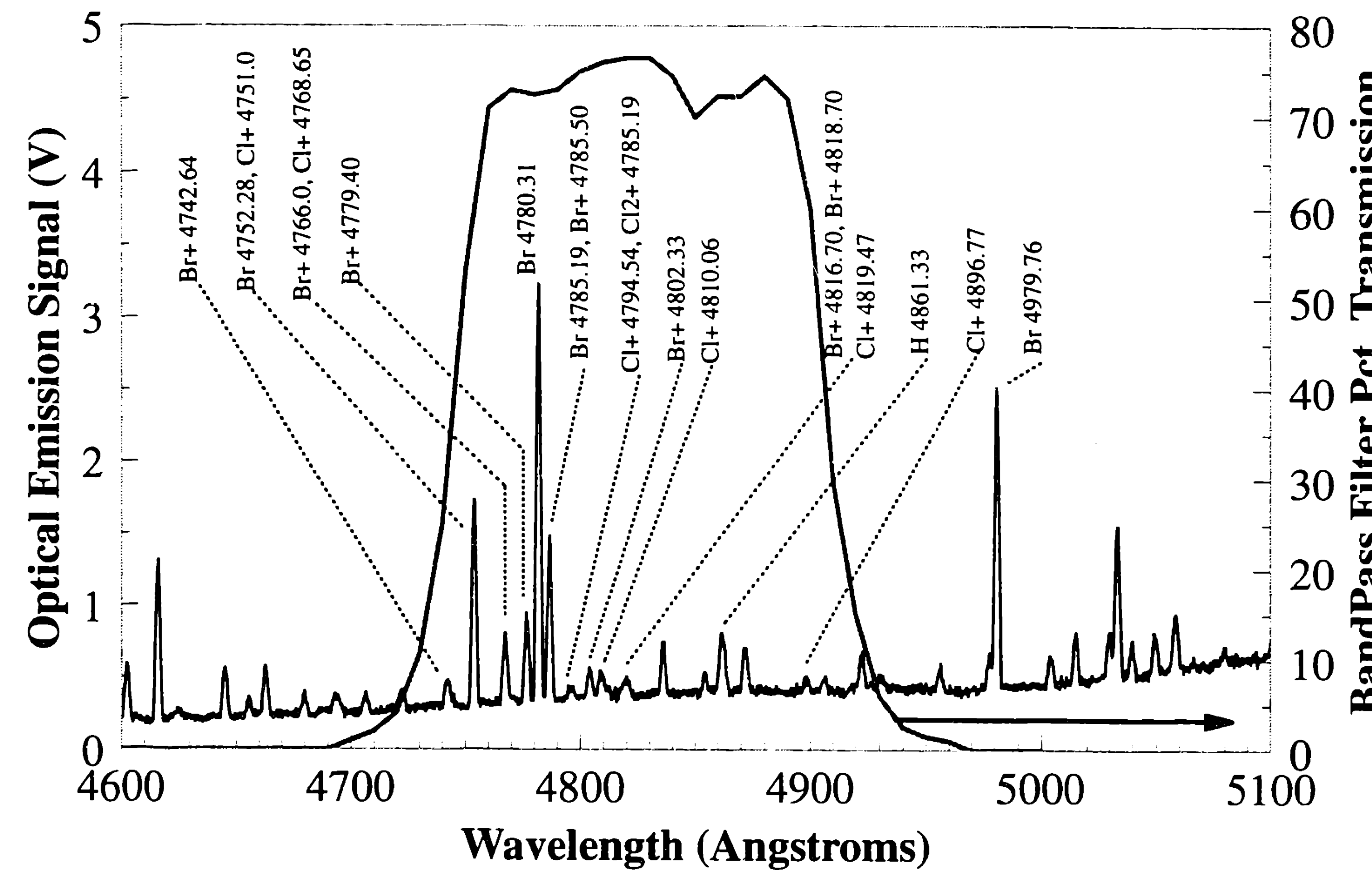


Figure 5-4B: Plasma Optical Emission Intensity for HBr/Cl2 Plasma Etching. Also shown is the Optical Band Pass Filter Percent Transmission for the 480 nm Filter Used in FWI.

not used to view an entire wafer due to the complexity in expanding a laser beam to illuminate a 100-mm diameter wafer.

5.2.3 Plasma Etcher and Magnetic Field Synchronization

Etching uniformity was measured on an Applied Materials Precision 5000 Plasma Etcher (AME-5000) donated by Applied Materials using both OEI and LI with a CCD camera as the detector. The AME-5000 operates in a MERIE (Magnetically Enhanced Reactive Ion Etching) configuration in which a rotating magnetic field is oriented so that the field lines are parallel to the cathode surface (perpendicular to the electric field). The magnetic field serves to increase plasma density by decreasing the electron loss rate coefficient. The field rotates to increase circumferential etching uniformity. This rotating field modulates the plasma intensity, thereby complicating the interpretation of the interferometry signal which is also modulated due to the etching of a thin film.

The effect of the rotating magnetic field was removed by synchronizing CCD image acquisition with the rotation of the magnetic field; this insures that data was always taken at the same phase of the magnetic field rotation. This effectively locked the field with respect to the data acquisition. There are spatial variations in plasma emission due to the field, but the temporal variations were removed from the collected data. Synchronization of data acquisition was accomplished using a Hall effect magnetic field sensor (MICRO SWITCH, Freeport IL, #SS21PE) which was interfaced to the personal computer using a Data Translation DT2817 Digital Input/Output card. Details are given in the Appendix.

5.2.4 Software for Data Analysis

Software was written using Microsoft C 7.0 to both acquire and analyze CCD images (see the Appendix for the actual code). CCD images were collected on a 31-megabyte RAM disk to maximize the data collection rates. One image (non-interlaced) occupied 186,912 bytes (766 x 244 bytes + 8 bytes for time stamp). The 33-Mhz 80386 PC achieved a maximum sampling rate (CCD read and data save) of 1.46 Hz; 169 complete images (representing about 2 minutes of etching at the maximum sampling rate) could be saved on the RAM disk.

Analysis regions included the entire wafer, an entire die, a line on the wafer and/or individual points. The analysis software displayed an image of the wafer, and then the regions to be analyzed were chosen from this image using a mouse-based interface. The series of images from an experiment were cycled through and the raw interferometry signal at each analysis point was determined. For small numbers of analysis points (less than 10), the software also computed the etching rate for these points. For a large number of analysis points, for example, a complete wafer map with over 700 analysis points, the etching rates were determined using a MIPS M120/5. The etching rate could be calculated for every pixel on the CCD corresponding to a location on the wafer surface.

5.2.5 Etching Rate Calculation

The etching rate was calculated from Equation 5-6, with Δd_{film} being calculated

from Equation 5-3. The angular correction factor from Equation 5-5 was calculated at each analysis point. This correction was very small; the maximum correction factor was less than 0.2% for polysilicon and less than 0.9% for silicon dioxide for a 100-mm wafer with the CCD system used in these experiments. Polysilicon's large index of refraction (3.42 versus 1.5 for silicon dioxide) results in a smaller correction factor because light is bent more towards the surface normal.

The time between adjacent extrema (maxima or minima) in the CCD interferometry signal was determined using a temporal Fast Fourier Transform (FFT) analysis with a decimation-in-time algorithm [Press et al., 1988]. The FFT algorithm required the number of data points to be an integral power of two; thus, the interferometry signal was zero padded to increase the size of the data set to meet the FFT algorithm's requirements. Further zero padding was used to increase the number of Fourier components computed in the spectrum (allowing greater resolution of the peak frequency) at the expense of additional computation time and memory usage. The etching rate was calculated directly from the FFT as

$$ER = \frac{\lambda}{2\ n_2\ \cos(\theta_2)} * SR * f_{max} \quad \textbf{(5-8)}$$

where SR is the sampling rate for the CCD and f_{max} is the dimensionless frequency at which the FFT amplitude is a maximum. The FFT frequency is nondimensionalized such that the transform is evaluated between frequencies of 0 and 0.5 (the Nyquist frequency).

The effects of zero padding the FFT are visible in Figure 5-5, where the calculated etching rate as a function of position is plotted for two FFT sizes, one with 512

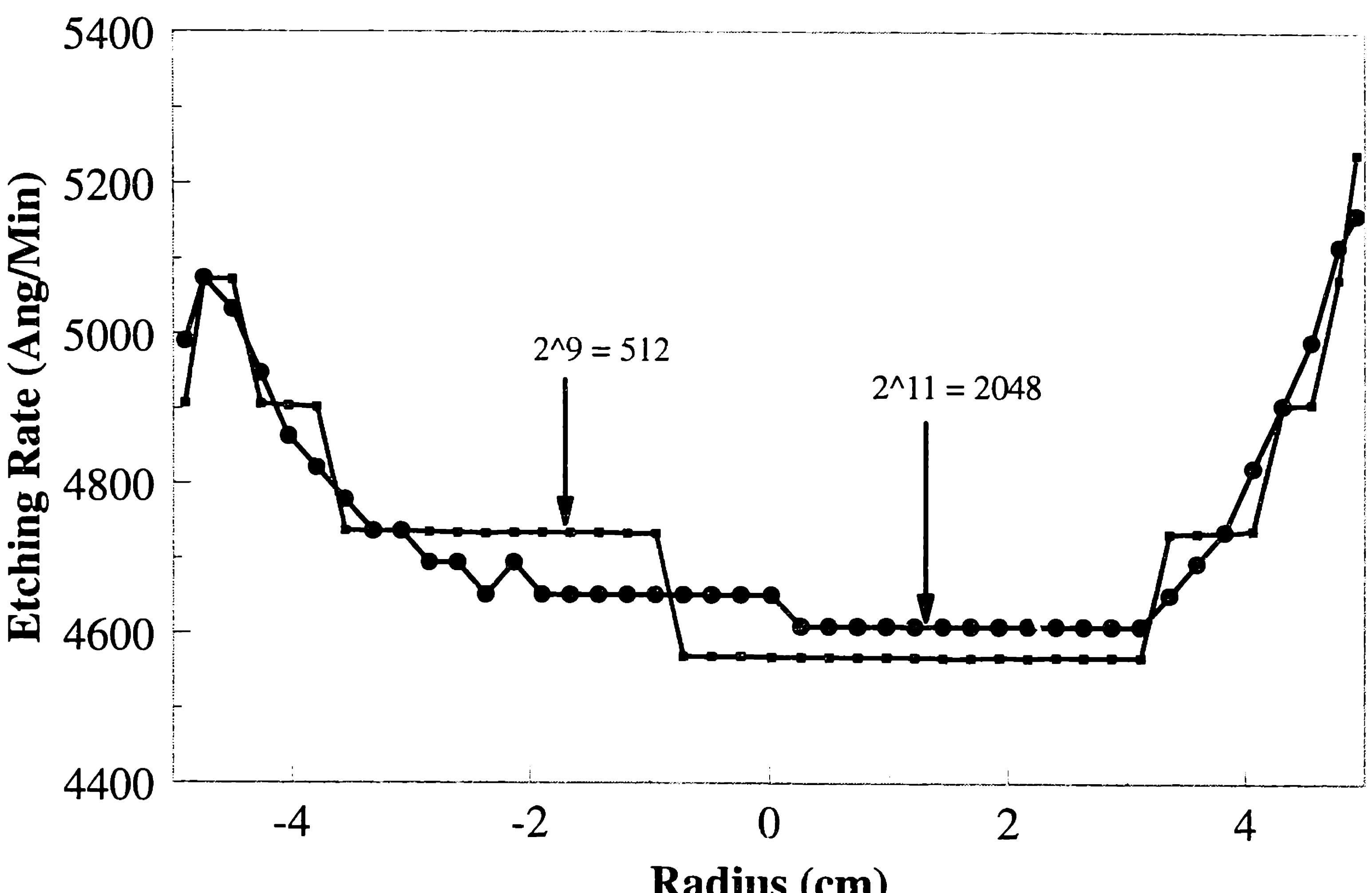


Figure 5-5: Effect of FFT Order on Calculated Etching Rate

(2^9)elements and on with 2048 (2^{11}) elements. Increasing the size of the FFT by zero padding leads to finer resolution of the etching rate. The curve for 2048 elements is smoother than the more coarsely resolved curve for 512 elements.

5.3 Results

Etching experiments were conducted using 5000Å of undoped polysilicon over 1000Å of silicon dioxide on 100-mm wafers. Both blanket polysilicon wafers and wafers patterned with 10,000Å of KTI820 photoresist were used. Etching was done on an AME-5000 using HBr and Cl_2 at 10 and 30 SCCM, respectively. Pressure was 100 mTorr; power was nominally 250 Watts, measured at the generator. The cathode temperature was 20°C, except where noted. Magnetic fields of 0 and 75 Gauss were used. A magnetic field rotation frequency of 1 Hz was used.

5.3.1 Measurement Noise and Comparison to Other Techniques

The signal-to-noise ratio in the observed CCD signals was measured by observing a non-eroding region of the process chamber during an etching run (Figure 5-6). The standard deviation of the observed CCD signal was 0.51 out of 256, or 0.2% of the full scale signal. Noise in the CCD signal (Figures 5-9A, 5-10 and 5-11A) is thought to be caused primarily by light scattering within the etching chamber and not from the CCD camera or plasma fluctuations. Light which is scattered from the wafer at any time before entering the collection optics can exhibit temporal fluctuations caused by thin film

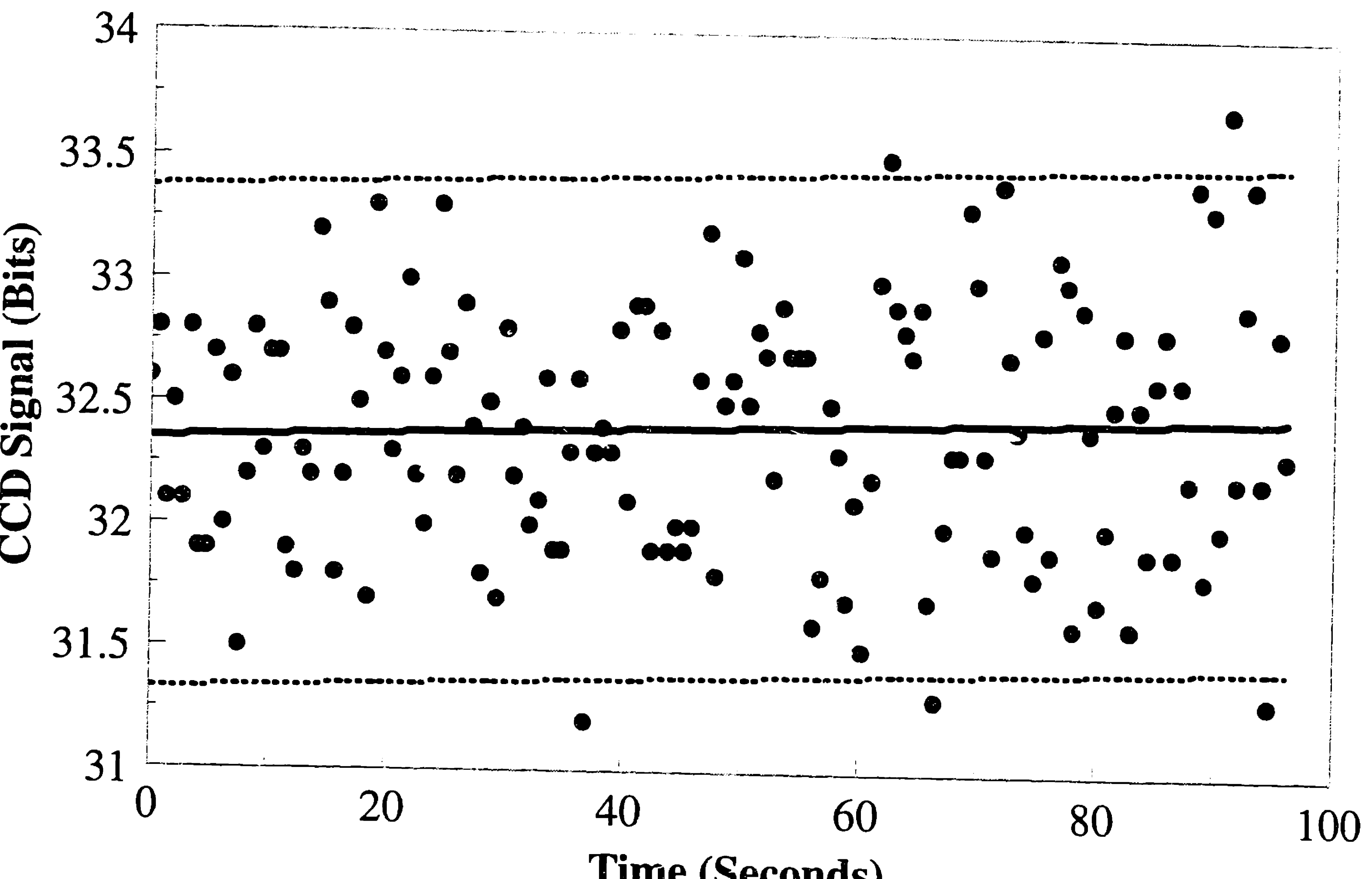


Figure 5-6 : CCD Signal for a Non-Etching Part of the Chamber. Shown are Data Points, the Best Fit Line, and the $\mp 2\sigma$ Lines.

interferometry and off-angle scattering from non-planar surface features.

Laser illumination, which eliminates much of the multiple scattering pathways, exhibits a much better signal-to-noise ratio and dynamic range as can be seen in Figure 5-7A where the CCD signals for OEI at 753.4 nm and for LI at 632.8 nm are plotted. The different periodicity in the two signals is due to the different wavelengths used, with the shorter wavelength exhibiting a shorter period. The OEI signal shows an increase in signal intensity near endpoint due to the increase in the optical emission intensity when clearing from polysilicon to silicon dioxide. The signal for LI using a photodiode detector is also plotted in Figure 5-7A. This compares very well with LI using the CCD. Etching rates determined with the CCD (OEI or LI) and with the photodiode (LI) were always within six percent of each other, and often much closer. The difference was mainly caused by the interpretation of the waveforms by FFT analysis.

The observed CCD signal often showed a linear increase in absolute signal intensity with time. This same effect has also been observed in conventional laser interferometry and in interferometry simulations [Boning *et al.*, 1994]. This is due to the finite (non-zero) absorption coefficient of silicon in the visible wavelengths. As the polysilicon film thins due to etching, the remaining thinner film reflects more light. A more prominent effect of signal absorption is expected towards lower wavelengths because the absorption coefficient is larger at 632.8 nm than at 753.4 nm (Table 5-1). This is verified in Figure 5-7A where the signal rise is greater at the lower wavelength. This effect is in contrast to phenomenon such as surface roughening of polysilicon during etching which is known to decrease the signal intensity with time.

Polysilicon and photoresist interferometry signals are shown in Figure 5-7B for

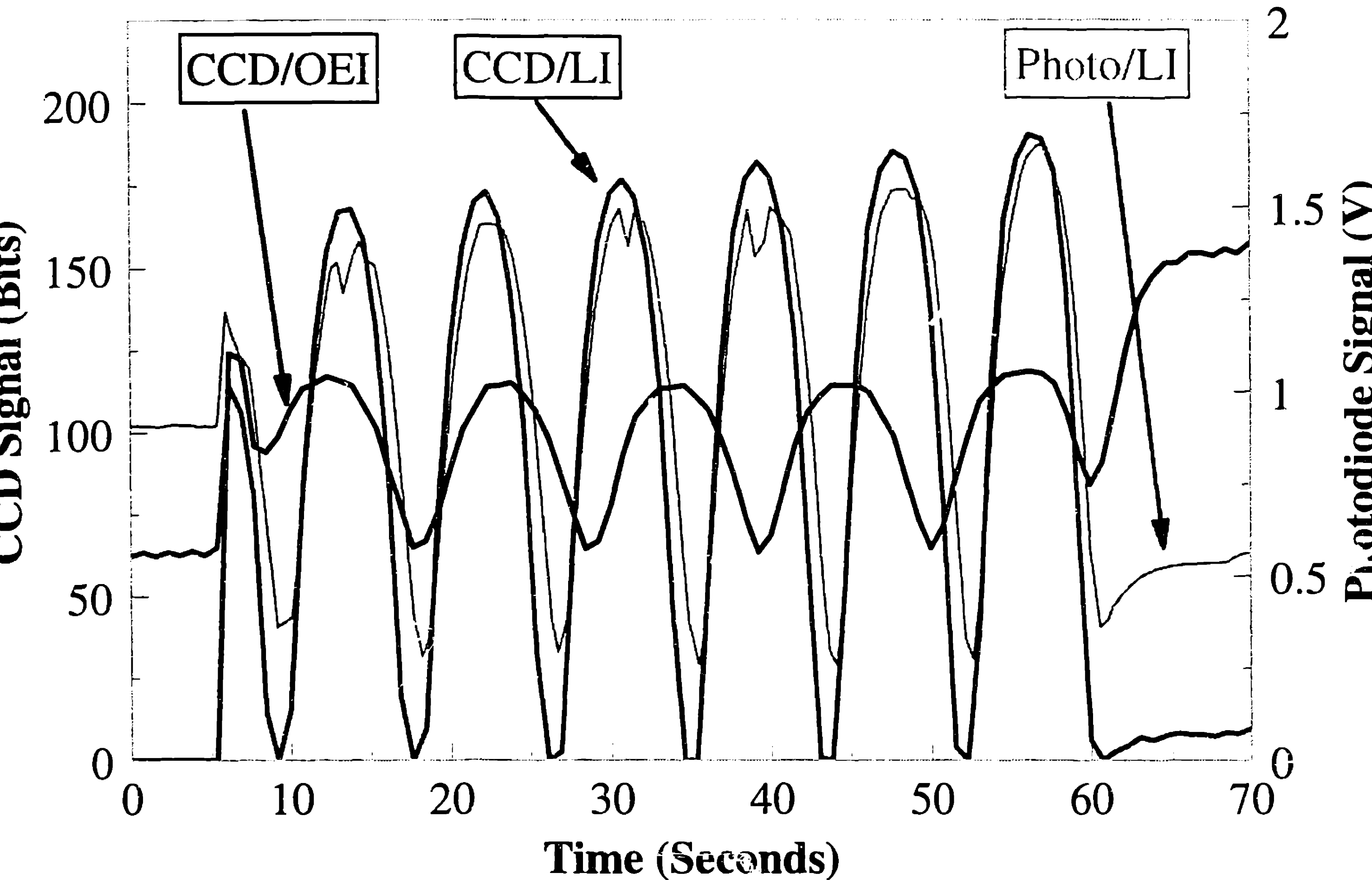


Figure 5-7A: Compaision of Interferometry Signals at the Same Location on the Wafer for Optical Emission Interferometry Using the CCD, and for Laser Interferometry Using the CCD and a Photodiode.

OEI at 480 nm. The polysilicon shows increased periodicity compared to 632.8 nm and 753.4 nm (Figure 5-7A) because of the shorter wavelength used. The absorption coefficient of polysilicon is an order of magnitude higher at 480 nm than at 753.4 nm. This results in the greatly reduced signal intensity for a thicker polysilicon film (small times on Figure 5-7B). The modeled signal is also shown on this graph. In order to match the small signal observed, a value of *k* for polysilicon four times that of single crystal silicon was used.

To reduce the affect of linear drifts in the signal, the data were fit to a straight line; this line was subsequently subtracted from the observed CCD signal. If this was not done, the FFT spectrum was sometimes dominated by the linear rise in the CCD signal and the etching rate could not be determined. As this calculation did not change the etching rates if the linear rise was not present, all of the FFT spectra (and etching rates calculated from them) presented here have been corrected for this linear component.

The etching rate determined by OEI with the CCD was compared to that determined with a Nanospec/AFT 200 (an instrument that determines film thickness by measuring optical reflection as a function of wavelength). To do this, the polysilicon film thickness was first measured at 12 sights per wafer with the Nanospec. The wafers were then partially etched and the film thickness measured again. The etching rates calculated by the two methods are plotted in Figure 5-8 along with the best-fit line (Solid) and the 45-degree line (Dashed). The etching rate determined with the CCD was always higher than that calculated with the Nanospec, with a greater difference at higher etching rates. The difference in the two methods may be due to: (1) the assumed etching time used to calculate the etching rate (ER = Δthickness / time); (2) failure to measure the same wafer

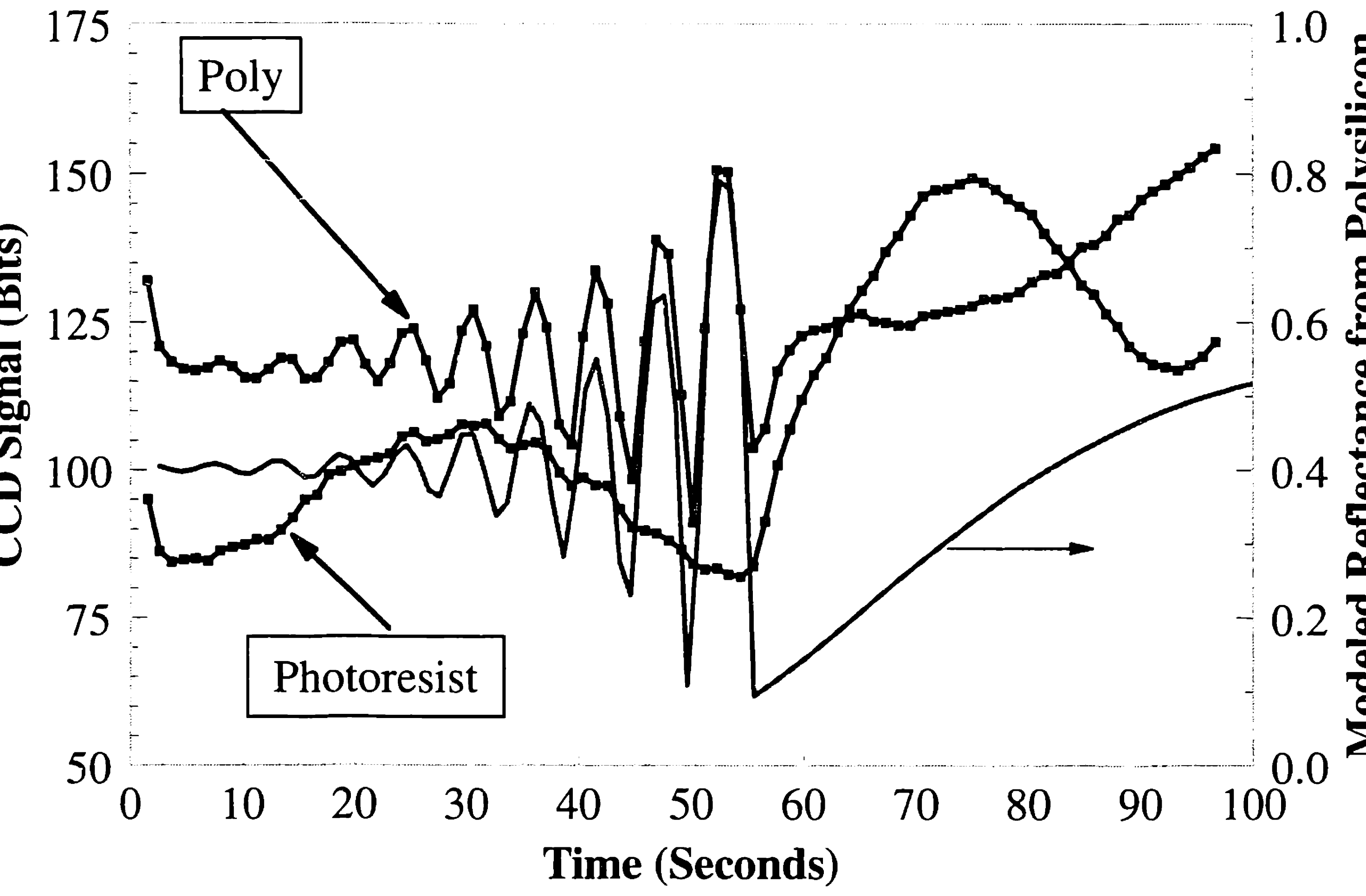


Figure 5-7B: Polysilicon and Photoresist Interferometry Signals at 480 nm. High Absorption Coefficient of Polysilicon Yields Weak Signal for Thick (5000 Å) Films. Also shown is modeled reflectivity of Poly.

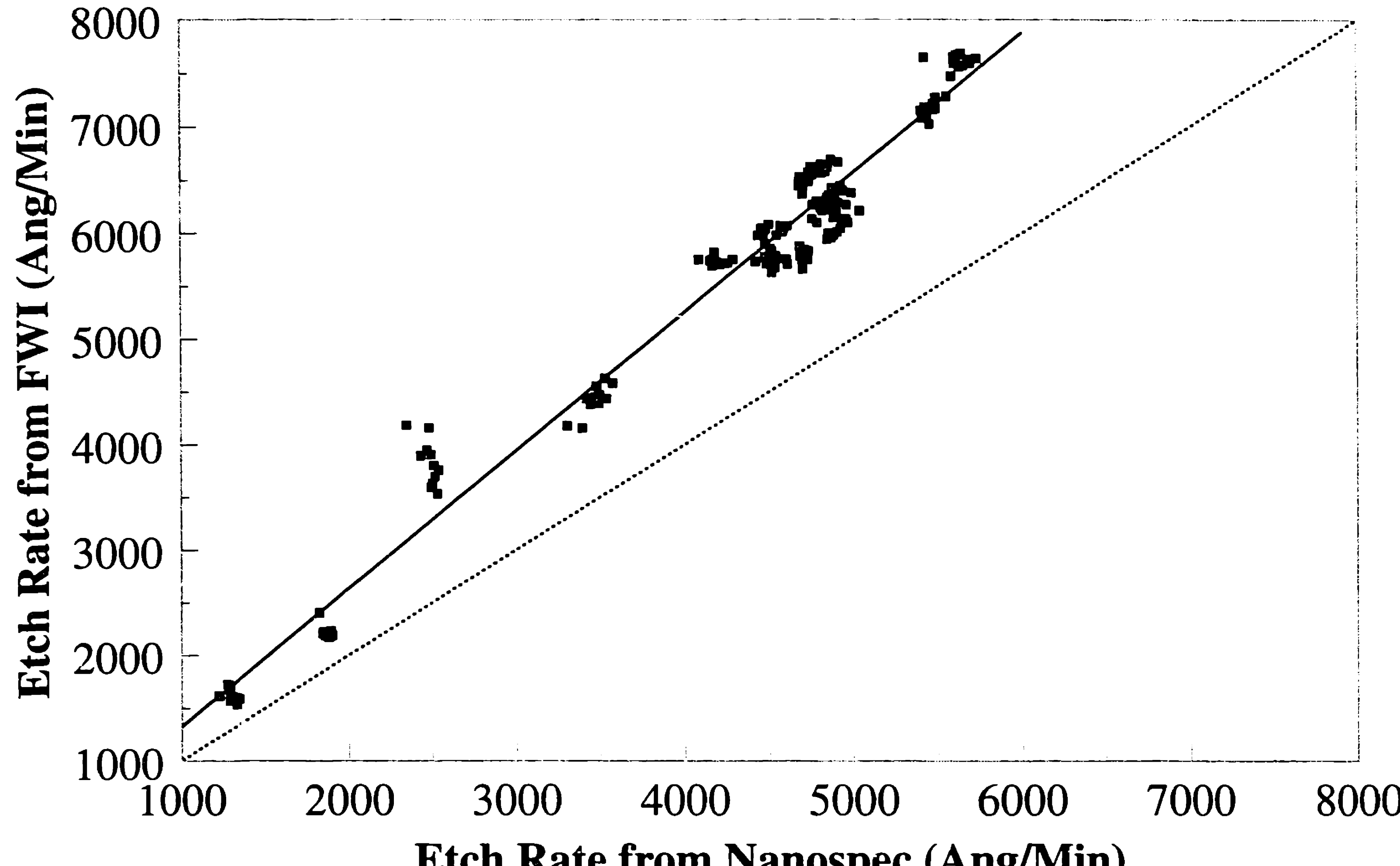


Figure 5-8: Comparison of Etching Rates Determined With FWI and With a Nanospec/AFT 200

site with the Nanospec before and after etching due to manual positioning for measurements; or (3) the use of an assumed average index of refraction for poly (3.0+0.0i) by the Nanospec over a wide wavelength range (480-800 nm). Earlier work comparing etch depths determined with a Dektak surface profilometer and with a Nanospec indicated that the index of refraction being used on the Nanospec may not be correct. More important than the absolute film etching rate is the relative film etching rate across a wafer (uniformity) which will be discussed in Section 5.3.6.

5.3.2 Analysis Area Effects

The number of pixels used in the analysis (size of analysis area) affected only the appearance of the time-domain interferometry signal. Shown in Figure 5-9A is the observed CCD signal for patterned polysilicon etching for a single pixel and for a 9-pixel average, centered about the location of the single pixel. The frequency-domain content was essentially unchanged by spatial averaging except at very high frequencies, which did not change the etching rate computation. Shown in Figure 5-9B is the FFT spectrum of the signals in Figure 5-9A. The 9-pixel average shows a very slight shift in the location of the maximum, which changed the etching rate by 1.1 percent.

Etching of unpatterned polysilicon showed much better signal-to-noise ratio and dynamic range than did patterned polysilicon. Etching rate variations with the size of the analysis area were vanishingly small. No significant deviations between the single-pixel and the 9-pixel analyses were seen. Figure 5-10 is the observed CCD signal for unpatterned polysilicon etching for a single pixel and for a 9-pixel average, centered

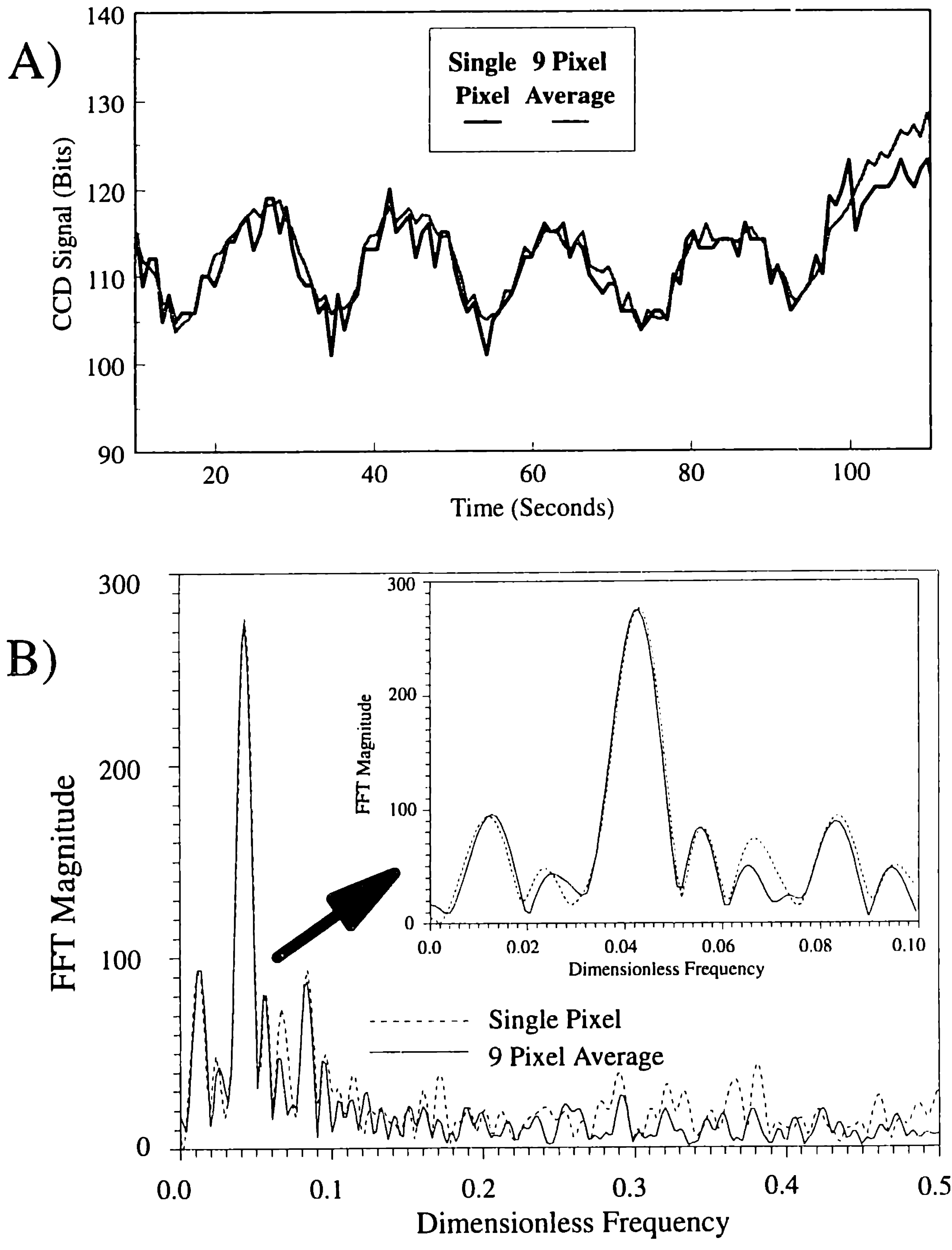


Figure 5-9: Observed CCD Signal and FFT Magnitude for the Etching of Patterned Polysilicon.

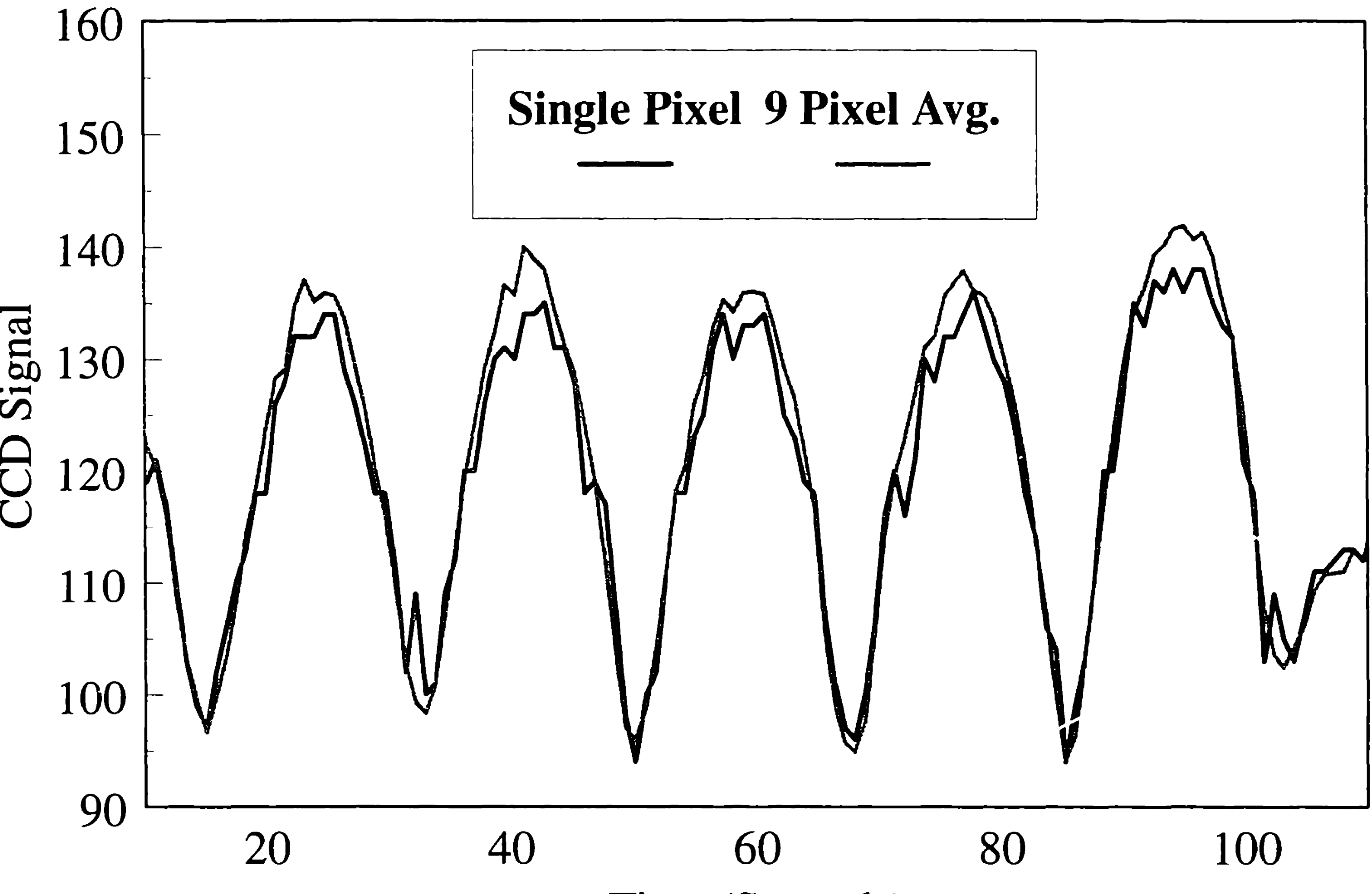


Figure 5-10: CCD Signal for the Etching of Unpatterned Polysilicon

about the location of the single pixel. Comparing this to patterned polysilicon etching in Figure 5-9A, better signal to noise ratio and better dynamic range are evident.

5.3.3 Observation of Pattern Dependent Etching

The interferometry signal from a polysilicon etch over a 3x3 pixel area (approximately 450x1050 μm) is plotted in Figure 5-11A for both a bare polysilicon region and a region with an array of holes in the resist masking the polysilicon; both regions are located on the same die. A different periodicity is observed for the two different regions indicating a difference in etching rate. Figure 5-11B shows the magnitude of the FFT for these two signals. Etching rates were determined to be 2960 Å/min for the bare polysilicon and 2475 Å/min for the polysilicon hole region. This observed difference in etching rates is a direct observation of pattern dependent etching, where the etching rate depends upon the local topography. This phenomena will be discussed in much greater detail in Chapter 8.

5.3.4 Interferometric Images

A time series of CCD images is shown in Figures 5-12A, 5-12B, 5-12C and 5-12D. Figure 5-12A is an image of the wafer 11 seconds after plasma ignition. Notice the uniform intensity of the polysilicon regions (dark dies lying along the diagonal in the lower left and upper right). Nonuniform etching can be seen at 48 seconds as is shown in Figure 5-12B. The nonuniformities are more visible at 80 seconds (Figure 5-12C) and

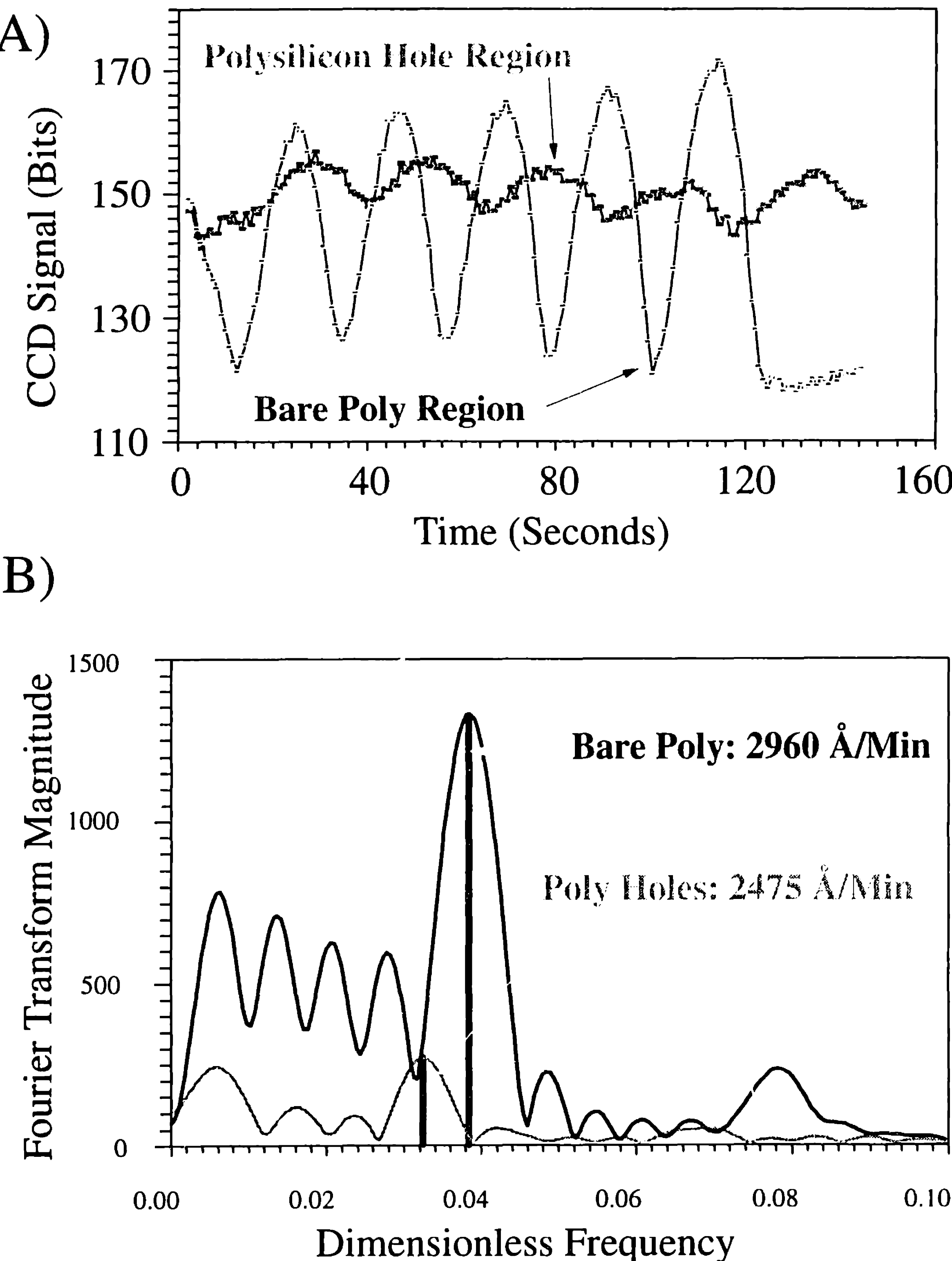


Figure 5-11: Observation of Pattern Dependent Etching Using the FWI Technique

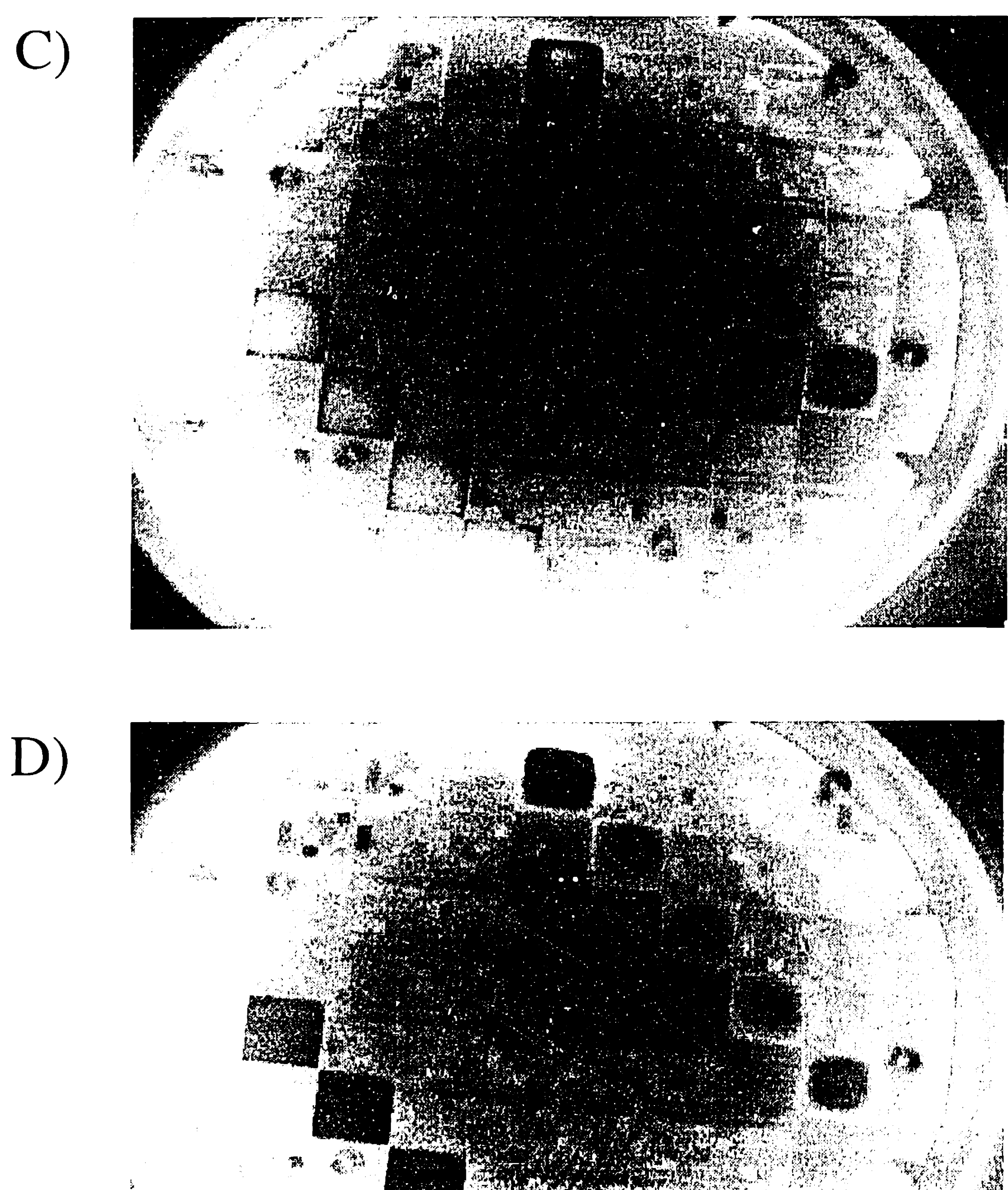


Figure 5-12: Time Sequence of CCD Images During Wafer Etch. C is 80 Seconds and D is 97 Seconds After Plasma Ignition.

finally, the polysilicon dies in the lower left have cleared at 97 seconds (Figure 5-12D), while the dies in the upper right are still etching. The patterns seen visually in these images are quantified in the next two Sections.

5.3.5 Uniformity Across a Die

Figure 5-13A shows a surface plot of etching rate across an unpatterned polysilicon die surrounded on three sides by photoresist (the polysilicon die located in the top row of Figure 5-12A). The etching rates shows a dome-like structure, with etching rate highest in the center and lowest near the edges. Figure 5-13B shows a contour plot of etching rate on the same die. The etching rate varies between 2950 Å/min at the center and about 2650 Å/min at the edge, a ten percent variation in uniformity across this die. The drop in etching rate near the edges may be due to microloading, where the etchants are consumed on the photoresist around the die. Alternatively, the decrease could also be caused by the deposition of photoresist products which passivate the polysilicon near the photoresist.

All the blanket polysilicon dies on the wafer showed a similar dome-shaped etching rate. The location of the maximum was typically at the center, but was skewed towards the edge of the wafer on dies located near an extreme edge of the wafer. This was caused by the combination of a dome-shaped etching rate across the die and the sharp increase in etching rate towards the edges of the wafer (discussed in Section 5.3.6).

5.3.6 Uniformity Across a Wafer

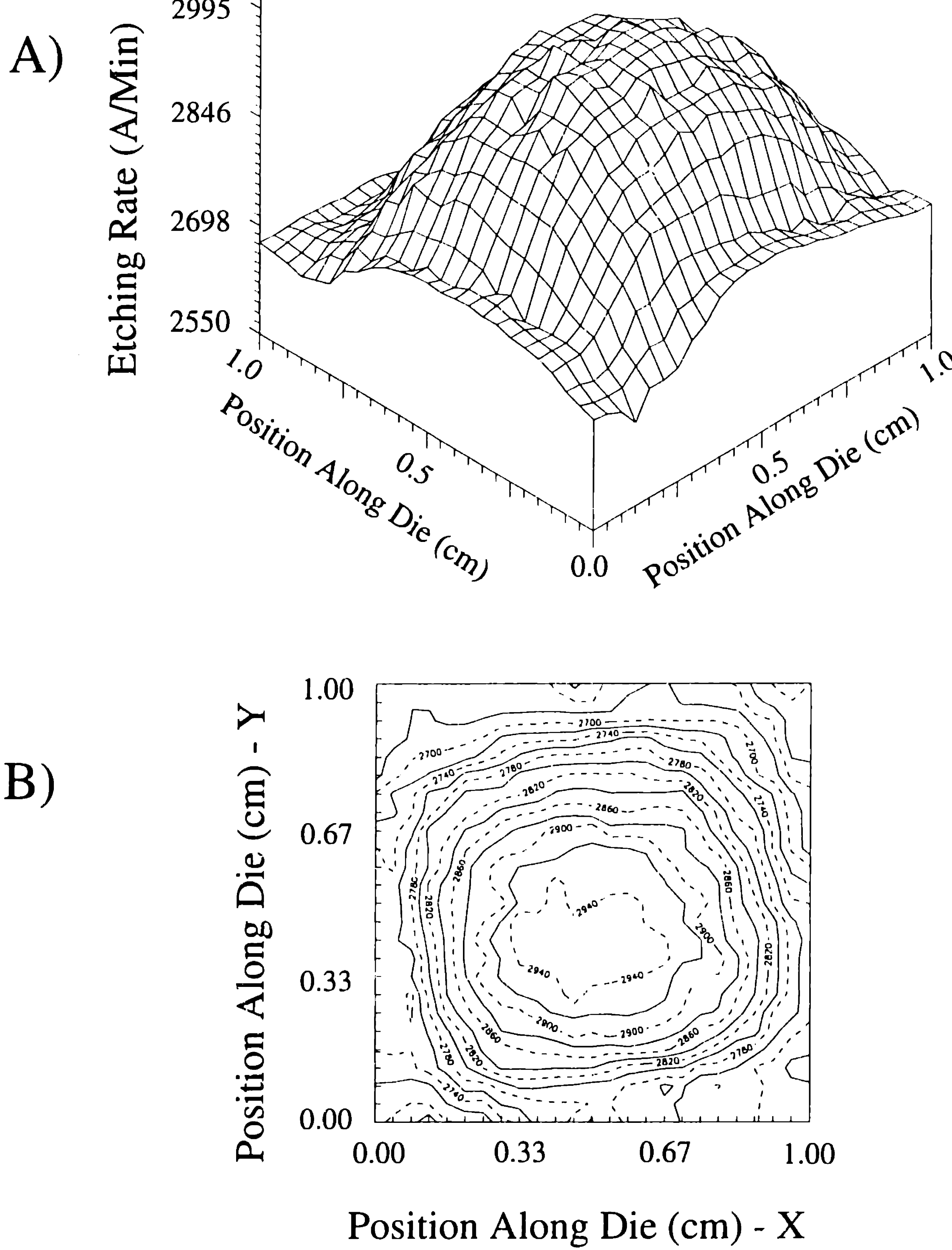


Figure 5-13: Etching Rate Across a Blank Polysilicon Die, Surrounded by Photoresist. Surface and Contour Plots.

Etching rates across an entire wafer can be measured, yielding uniformity. Figure 5-14 shows a surface plot of polysilicon etching rate across an unmasked 100-mm polysilicon-covered wafer etched using no magnetic field, while Figure 5-16 shows the same surface for a wafer etched with a 75 Gauss magnetic field revolving at 1 Hz. A large difference can be seen between the two responses. In the first case, without a magnetic field, the etching rate is lowest near the center, and increases radially outward. The actual location of the etching rate minimum is shifted off of center, at about -45 degrees; the load lock slit valve is located at about -90 degrees and the turbo pump port at +90 degrees. With the magnetic field turned on, the etching rate decreases radially outward. Also visible in these plots is the effect of the quartz clamp ring used to hold the wafer on the cathode; there is a large drop in the etching rate where the fingers on the clamp ring touch the wafer, as seen on the edge of these surface plots.

The clearing pattern seen in Figure 5-14, etching fastest at the edge and slowest at the center has been termed a "bullseye" pattern [Danner *et al.*, 1987]. Danner *et al.* [1987] attributed this pattern in BCl_3/Cl_2 etching of Al to the diffusion of reactants through a mass transport boundary layer. Giapis *et al.* [1990] also observed this "bullseye" clearing pattern in the etching of AlGas/AlAs with BCl_3/Cl_2 at 27°C. This nonuniformity could be decreased by lowering the wafer temperature to -40°C.

The "bullseye" pattern is typically caused by differences in the reactant transport to the edge and to the center of the wafer. The importance of this transport variation is determined by the rapidity of surface reaction with respect to the reactant transport to the wafer. This situation is characterized by a Damköhler number of the second kind, Da_{II}, which is the ratio of the characteristic reaction velocity (ϑ_r) to the characteristic diffusion

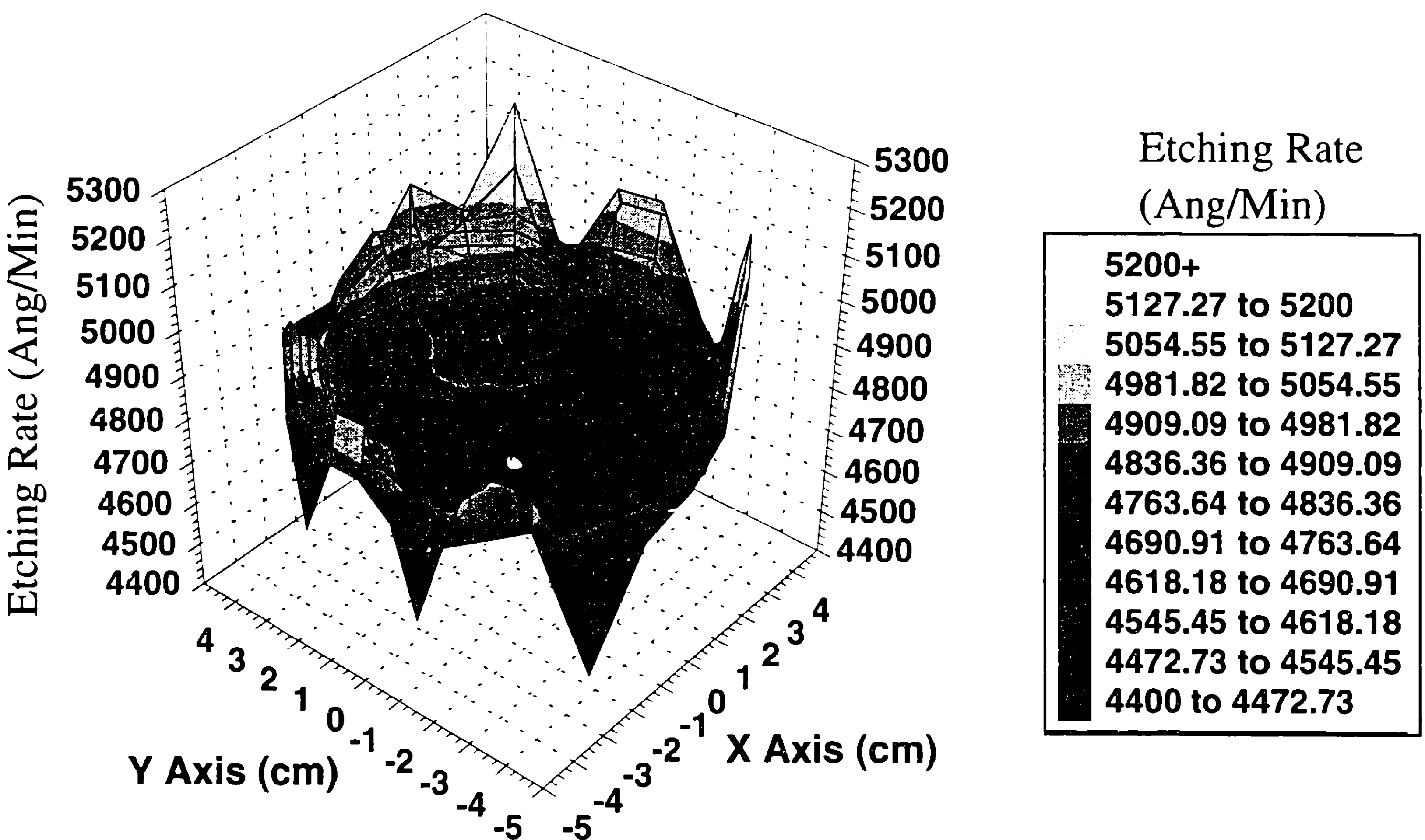


Figure 5-14: Etching Rates Across an Entire Wafer, Measured *in situ*. Magnetic Field Strength = 0 Gauss.

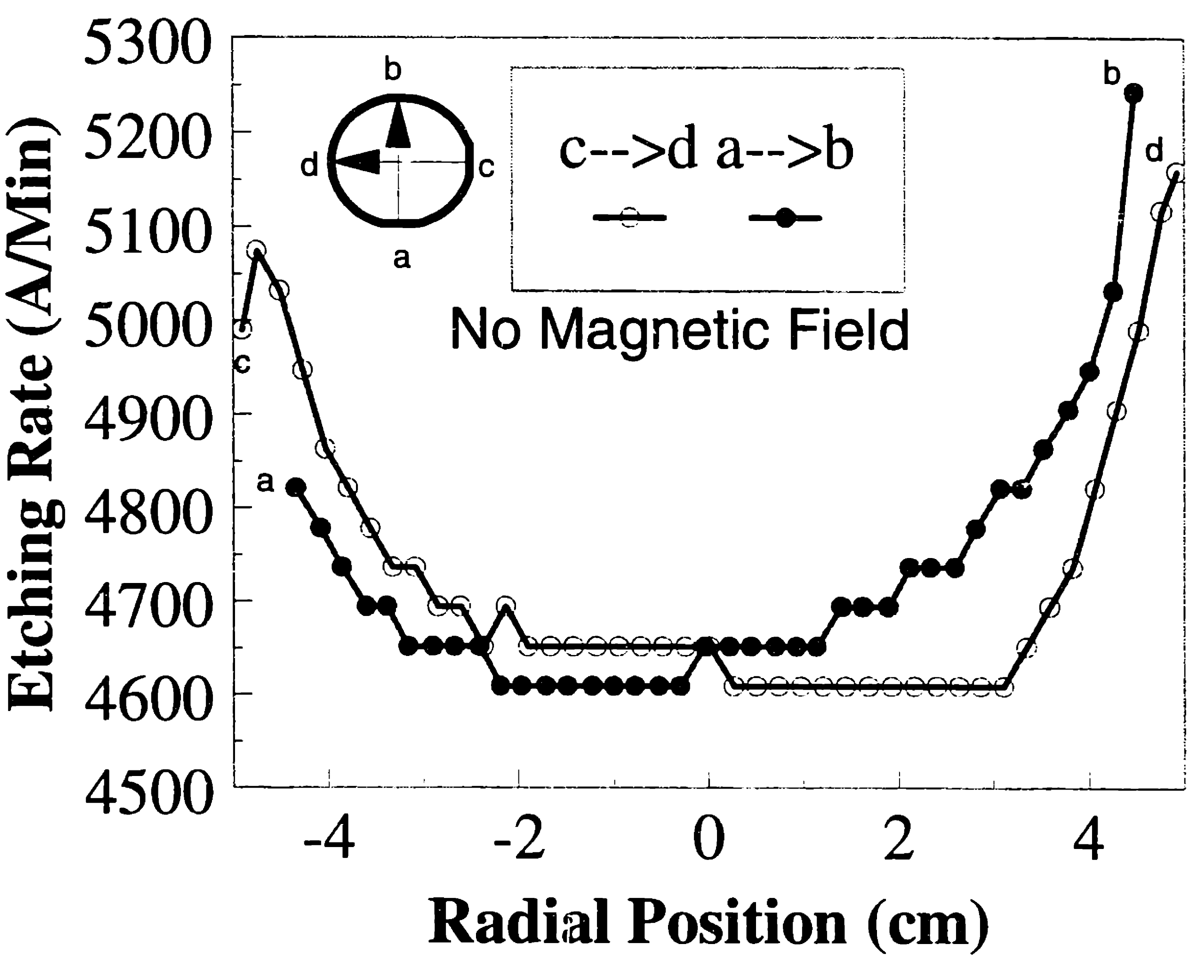


Figure 5-15: Etching Rates Across an Entire Wafer, Measured in situ. Magnetic Field Strength = 0 Gauss.

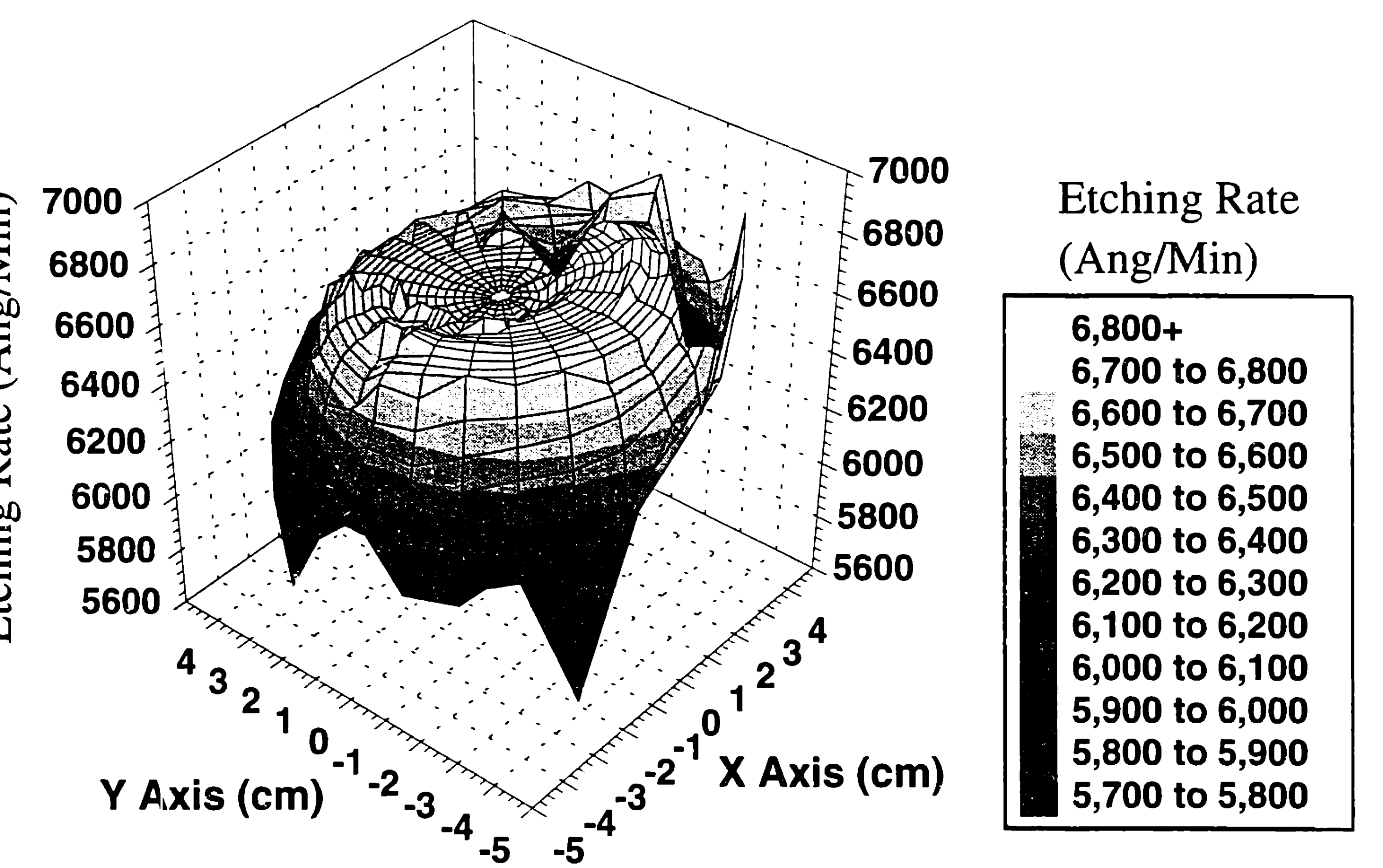


Figure 5-16: Etching Rates Across an Entire Wafer, Measured *in situ*. Magnetic Field Strength = 75 Gauss.

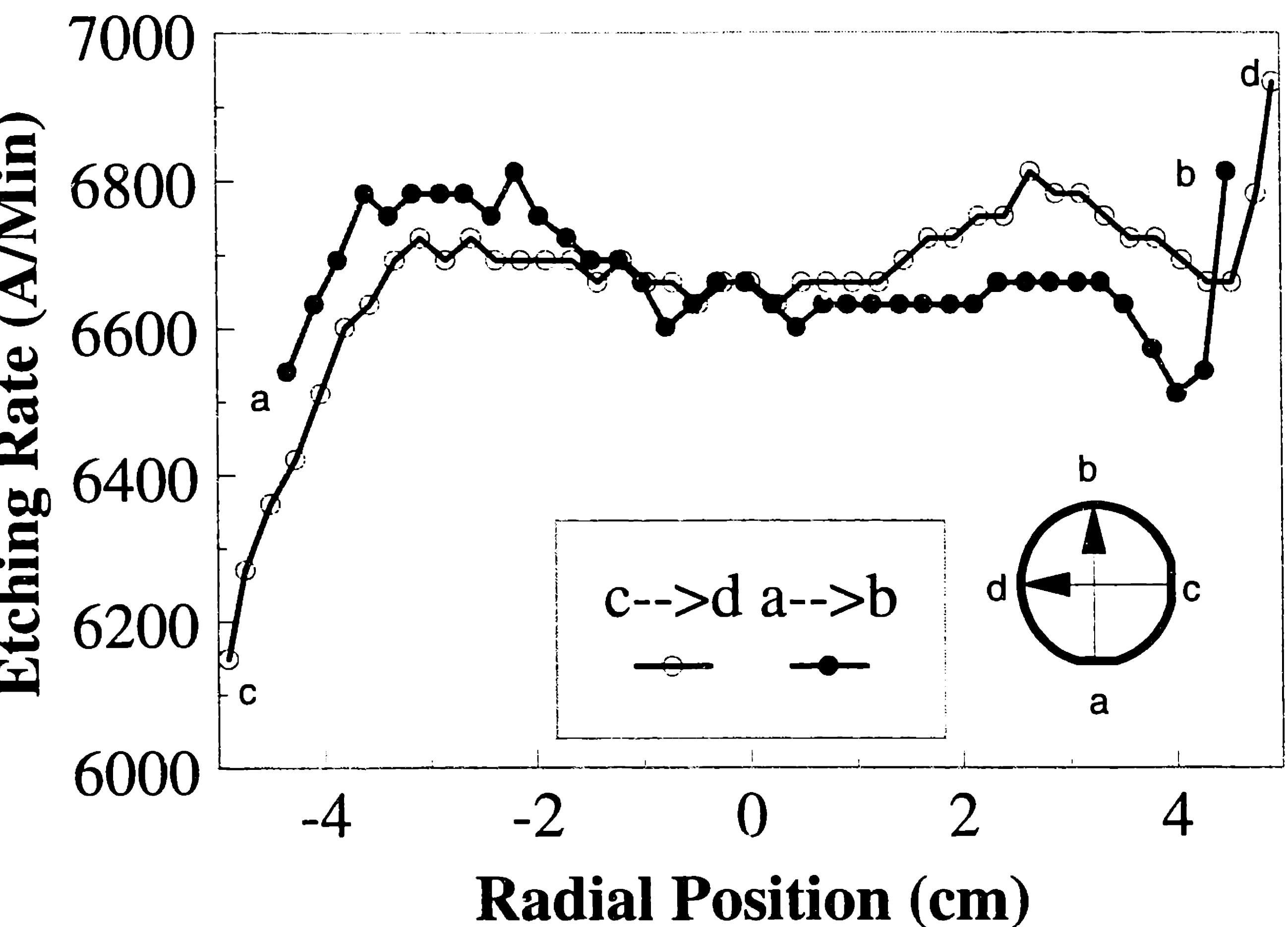


Figure 5-17: Etching Rates Across an Entire Wafer, Measured in situ. Magnetic Field Strength = 75 Gauss.

velocity (ϑ_D) and can be expressed as

$$Da_{II} = \frac{\vartheta_r}{\vartheta_D} = \frac{r}{D/l} = \frac{rl}{D} \qquad \textbf{(5-9)}$$

where l is a characteristic length for diffusion, D is the diffusivity of reactants (in the gas phase), and r is the surface reaction rate coefficient. Large values of Da_{II} indicate reactant transport is much slower than reaction, while small values indicate that reactant transport is much faster than reaction. The surface reaction rate coefficient depends exponentially on surface temperature, through the Arhenius behavior of the reaction rate coefficient. Decreasing the wafer temperature will decrease r and thus decrease ϑ_r and thus decrease Da_{II} which tends to move the etch process from a reactant transport limited regime to a surface reaction rate limited regime. The gas phase diffusivity of reactants will also decrease with temperature, but at a slower rate than the surface reaction rate coefficient. Bird *et al.* [1960] show that kinetic theory predicts that $D(T) \propto T^{3/2}$.

Plasma etching of undoped polysilicon in Cl_2 on the AME-5000 showed a very similar response to that of HBr/Cl_2. Reducing the cathode temperature from 20°C to -75°C, which yielded a wafer surface temperature of around -60°C during etching reduced the "bullseye" pattern and produced a radially more uniform etch as is shown in Figure 5-18, which is with a 75G magnetic field in pure Cl_2. The average etching rate decreased from around 6700 Å/Min at 20°C to around 5700 Å/Min at -75°C. The dramatic improvement of etch uniformity at -75°C indicates that the ion flux is fairly uniform, and that nonuniform etching at higher temperatures is due to nonuniform reactant transport. Cross-sectional views of etching rate as a function of position for cross-sections taken

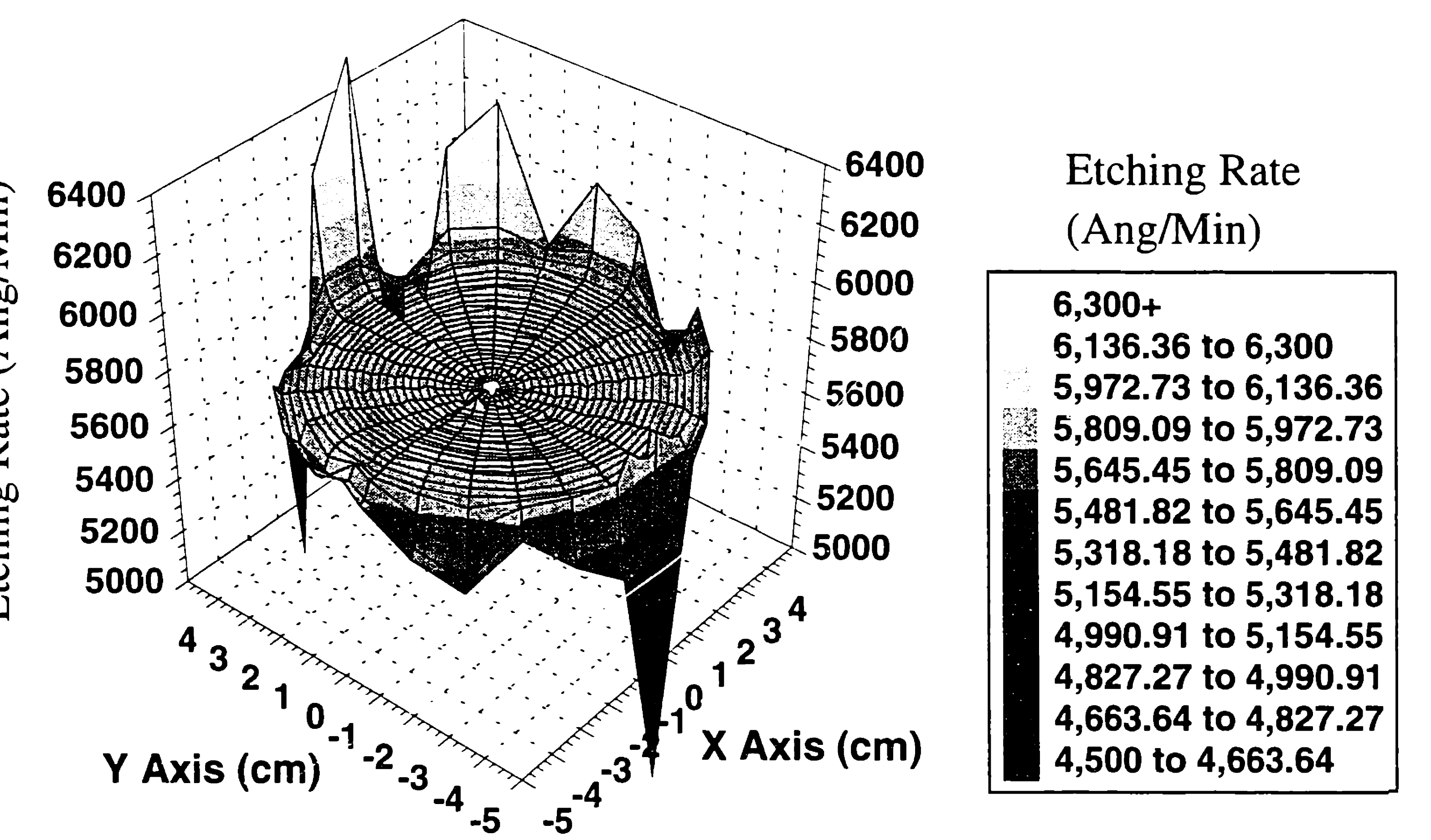


Figure 5-18: Etching Rates Across an Entire Wafer, Measured *in situ*. Magnetic Field Strength = 75 Gauss, Temperature = -75°C.

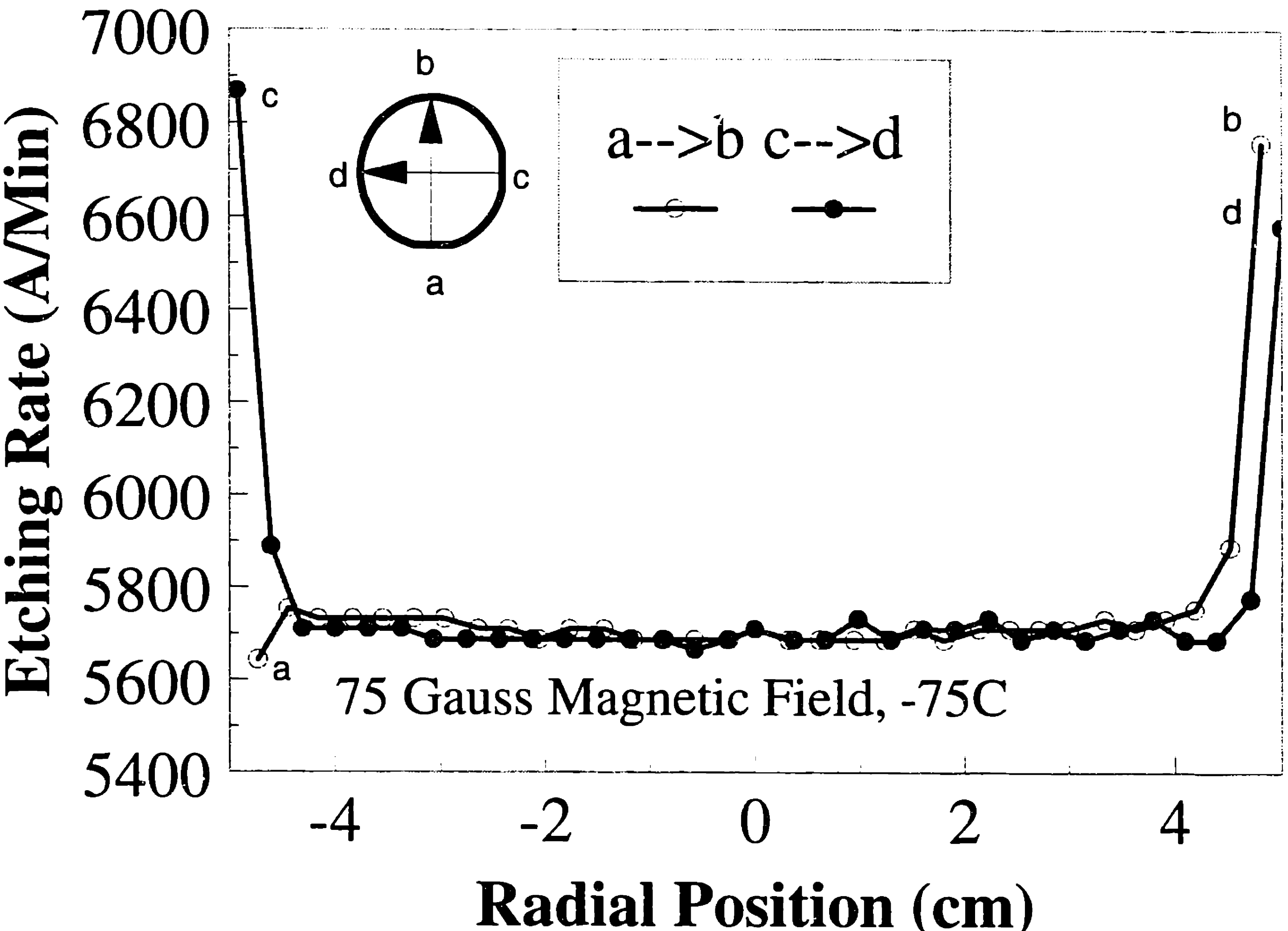


Figure 5-19: Etching Rates Across an Entire Wafer, Measured *in situ*. Magnetic Field Strength = 75 Gauss, Temperature = -75°C.

along the x and y axes of Figures 5-14, 5-16, and 5-18 are shown in figures 5-15, 5-17, and 5-19, respectively.

The change in the uniformity profile for polysilicon etching on the AME-5000 as magnetic field strength varies indicates that the "bullseye" pattern is being formed by differences in the plasma density across the wafer. The effect of magnetic field strength on radial nonuniformity across a patterned wafer is shown in Figure 5-20 for a nominal etching power of 375 Watts. Lines are plotted for uniformity determined using the CCD and using the Nanospec. Radial nonuniformity is calculated from the slope of the etching rate/radial position plot determined by calculating the etching rate for several azimuthal positions at several different radial positions. At a magnetic field of about 75 Gauss, etching with the least radial nonuniformity is obtained.

Radial nonuniformity was measured for 9 runs over several days time during other wafer processing using both the CCD and the Nanospec (Figure 5-21). The large jump in nonuniformity at run 5 for the CCD is due to the very short etch time used, which yielded only slightly more than 1 cycle of extrema on the interferometry signal, and made the calculation of etching rates less accurate. The difference in uniformity between the CCD and the Nanospec may be explained by some of the reasons listed in Section 5.3.1.

5.3.7 Effect of Film Thickness

Polysilicon (5000 Å) was etched with four different underlying oxide film thickness (100, 220, 440 and 1000 Å) and analyzed using FWI (Figure 5-22). The absolute signal intensity and dynamic range decreased as the oxide film thickness

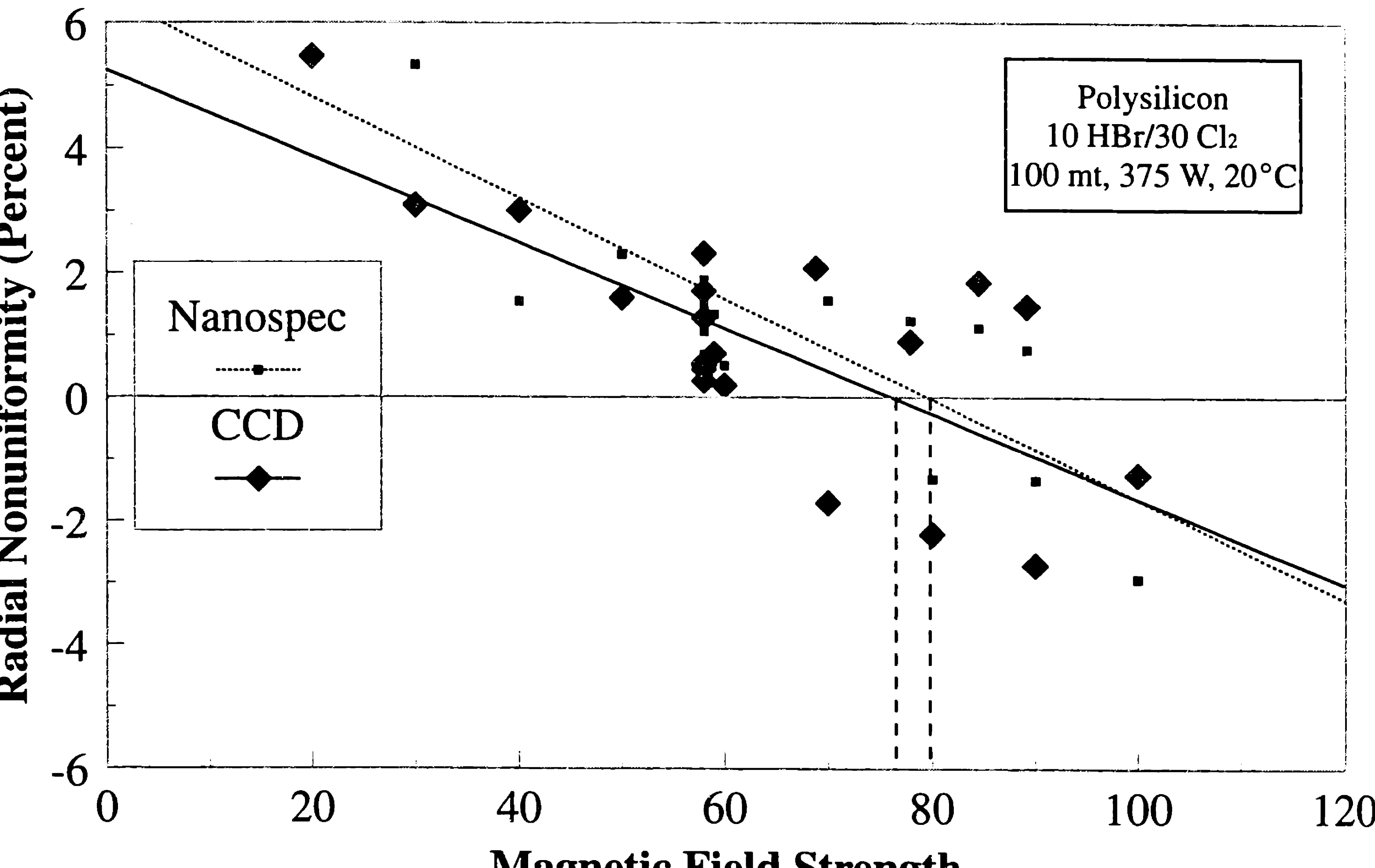


Figure 5-20: Effect of Magnetic Field Strength on Radial Uniformity for Undoped Polysilicon Etching at 375 Watts.

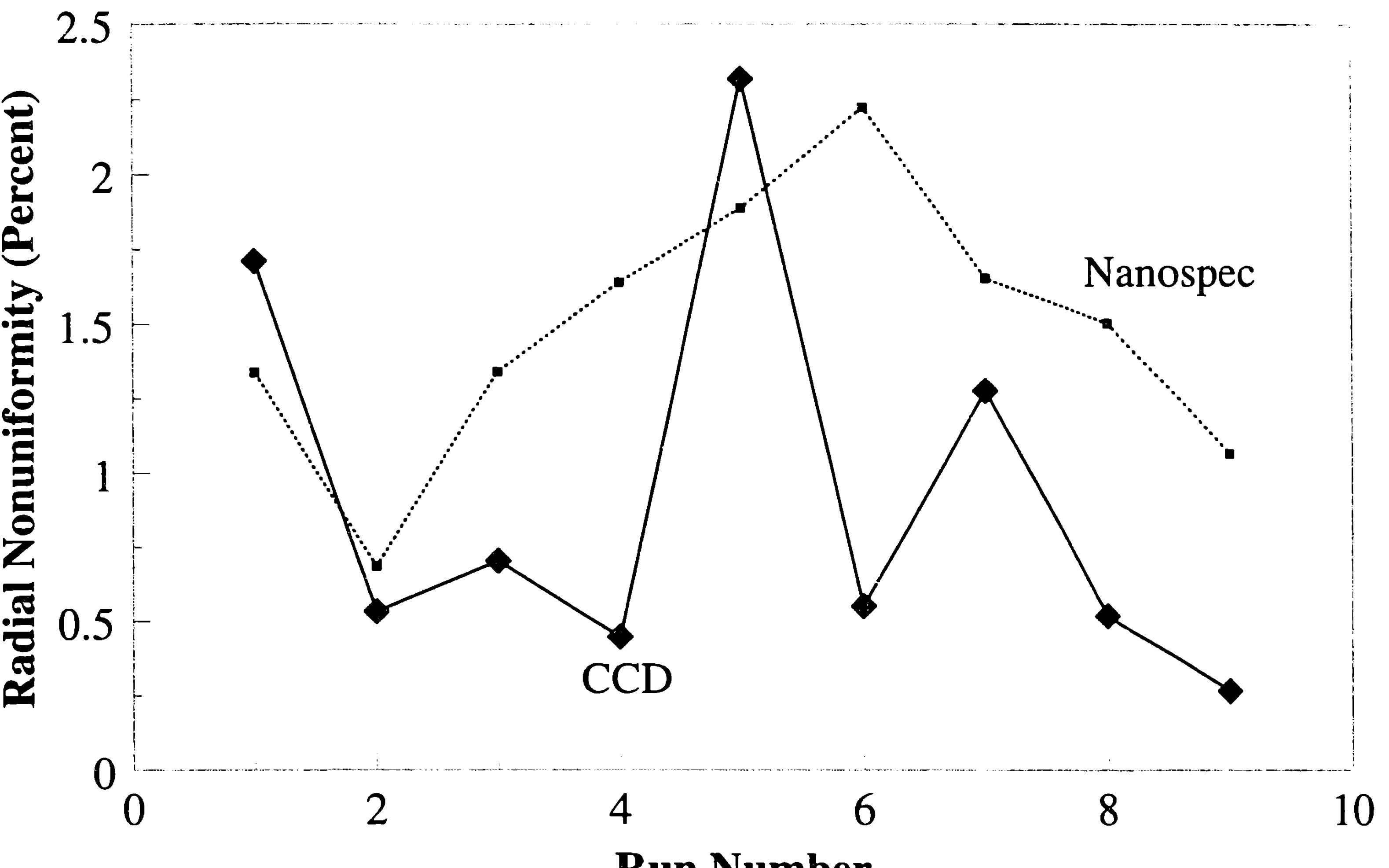


Figure 5-21: Radial Nonuniformity For Several Runs Over Several Days as Determined by CCD FWI and by Nanospec

decreased. This compares very well with the signal predicted using a simple optical reflectance model (Figure 5-23) [Anders, 1967]. Different end-point "signatures" are visible in this plot for the different thicknesses of the underlying oxide film. A model for the observed reflectivity aids in interpreting interferometry signals (i.e. Figure 5-22) because the correspondence of features in the observed signals to changes in film thickness can easily be seen in the model.

5.4 Determining End-Point Time and Uniformity

Etching uniformity was discussed in Section 5.3.6. Full wafer optical emission interferometry was compared to optical emission spectroscopy, laser interferometry, plasma electrical property measurement, and wafer temperature measurement for use as an end-point tool. Full wafer interferometry has the advantage of being a spatially resolved method. For the etching of both undoped and P-doped polysilicon in an Applied Materials Precision 5000 plasma etcher, full wafer interferometry consistently determined the start of wafer clearing before any other method. Furthermore, it was first to detect the completion of clearing. Full wafer interferometry is suited for use as an end-point diagnostic in manufacturing to control more tightly plasma etching processes.

5.4.1 Introduction to End-Point Determination Methods

It is important to stop the plasma etching process after the film being etched has

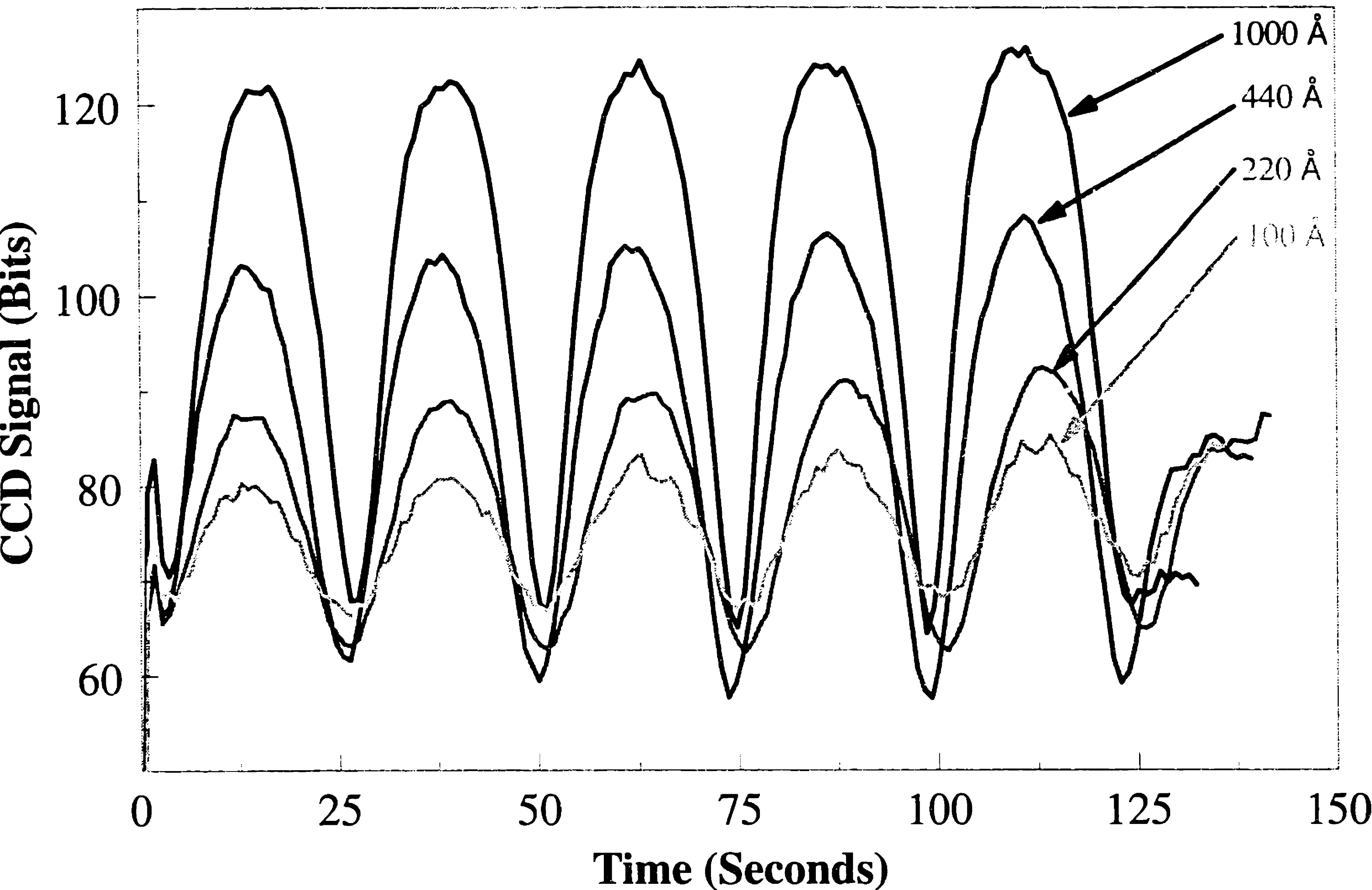


Figure 5-22: Effect of Underlying Oxide Film Thickness on FWI Signal for Unpatterned Polysilicon at Wafer Center.

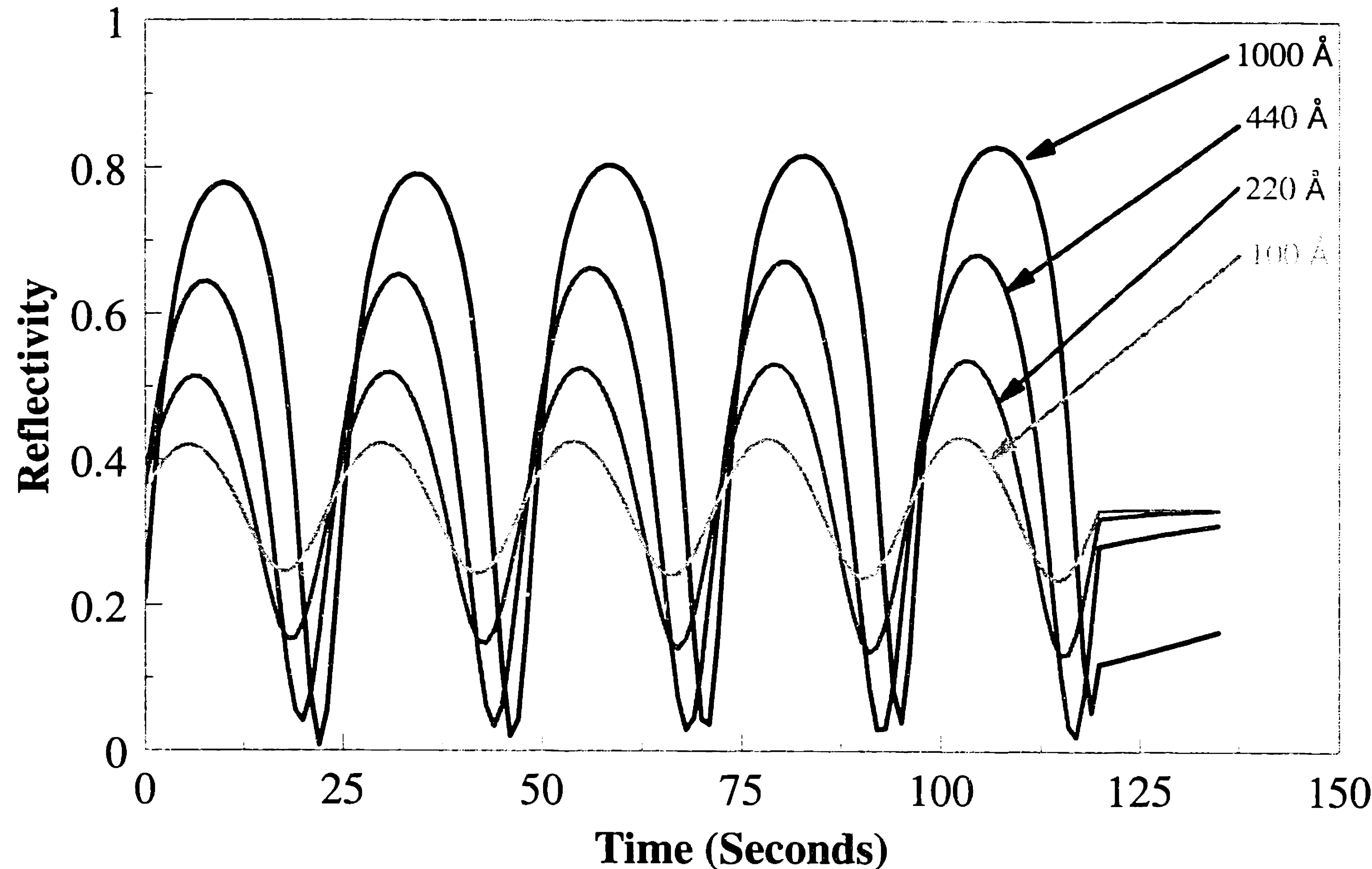


Figure 5-23: Model Results for Effect of Underlying Oxide Film Thickness on Reflectivity

been cleared (i.e., end-point is reached) to avoid damaging or removing an underlying film. An *in situ* diagnostic tool is necessary to control the plasma etching process; it must accurately determine the end-point time. Furthermore, it must also determine the end-point uniformity because the time required to reach end-point can vary as a function of position across a wafer.

A number of different techniques have been used to determine plasma etching end-point. These techniques monitor a change either in (1) a bulk plasma property, e.g. optical emission spectroscopy (OES) [Degenkolb et al., 1977],[Griffiths and Degenkolb, 1977],[Harshbarger et al., 1978], laser-induced fluorescence (LIF) [Selwyn, 1988], mass spectrometry [Raby, 1978],[Brown *et al.*, 1978], plasma impedance [Ukai and Hanazawa, 1979],[Patel *et al.*, 1992], D.C. bias voltage [McNevin *et al.*, 1993], pressure [Hitchman and Eichenberger, 1980], and Langmuir probe [Kawamoto and Hashimoto, 1979]; or in (2) a wafer surface property, e.g. laser interferometry (LI) [Busta *et al.*, 1981],[Sternheim and van Gelder, 1983], laser diffraction [Kleinknecht and Meier, 1978],[Braga *et al.*, 1983], optical emission interferometry (OEI) [Heinrich *et al.*, 1989],[Angell and Oehrlein, 1991], ellipsometry [Henck, 1992],[Henck *et al.*, 1993], and wafer temperature [Patel *et al.*, 1991]. Several comparisons of different end-point techniques have been published [Marcoux and Foo, 1981],[Roland *et al.*, 1985], and [Collot *et al.*, 1991]. Harshbarger [1984] has published a thorough discussion of plasma diagnostics and end-point detection.

An end-point algorithm is necessary to determine the end-point time from data obtained using one of the above mentioned end-point techniques. For example, in OES, both product and reactant species can be monitored. Typically, reactant species show a

signal increase at end-point and product species show a decrease at end-point, such as O and CO, respectively, during photoresist stripping [Degenkolb et al., 1976],[Griffiths and Degenkolb, 1977]. In another example, using laser interferometry/reflectometry, a change in film is signaled either by a change in periodicity (for a transparent/semi-transparent film) or by a change in reflectivity (for an opaque film). The number of possible algorithms for end-point determination is very large. One approach being used is to analyze data with an expert system both to determine end-point and for process diagnosis [Dolins et al., 1988].

Uniformity of etching is important because nonuniform etching necessitates the use of an over-etch step to completely clear the film from the wafer surface [Degenkolb and Griffiths, 1977]. Consequently, underlying films are exposed to plasma for different times depending on when the upper film clears. Thus, spatially-resolved etching information, which none of the above-mentioned end-point methods yields, is important. These techniques either average across the plasma bulk (as in OES or impedance) or sample a single point on the wafer surface (as in LI or ellipsometry). Previous researchers have acknowledged that the rate of change of an end-point signal at end-point is related to the uniformity of etching [Raby, 1978],[Patel *et al.*, 1992],[Johnson, 1980] but have been unable to quantify this relationship. Accurate knowledge of end-point across an entire wafer is necessary to more tightly control plasma etching processes, especially because etch-stop layers are becoming thinner (for example, stopping of a polysilicon gate etch on a sub-100-Å silicon dioxide layer) [Dalton *et al.*, 1993].

Full wafer interferometry (FWI) can determine spatially resolved etching rates and end-point times. By adding the spatial dimension to end-point determination, significant

improvement in process control is expected. In this section, we compare this new method for determining end-point with several other methods.

Experiments were conducted on an Applied Materials Precision 5000 plasma etcher (AME-5000), running either as a magnetically enhanced reactive ion etcher (MERIE) or as a reactive ion etcher (RIE) with 100-mm wafers. A number of diagnostic devices have been installed on this etcher. Figure 5-24 shows a schematic view of the etcher and diagnostics installed on it, which have been described in Chapter 1. They include: full wafer interferometry, photodiode laser interferometry, plasma electrical properties via a Comdel real power monitor, optical emission spectroscopy, and wafer temperature via a Luxtron 755 Fluoroptic® thermometer.

5.4.2 End-Point Experiments

Test wafers consisted of 5000-Å blanket films of polysilicon (undoped, or 13 Ω/□ phosphorous doped) over 1000 Å of silicon dioxide on 100-mm {100} n-type silicon wafers. Etching was done using a mixture of Cl_2 (30 SCCM) and HBr (10 SCCM) at 100 mTorr pressure. Power was 250 W as measured at the RF generator. The cathode temperature was 0°C, with 4 Torr of He backside cooling. A magnetic field strength of either 0 or 75 Gauss, rotating at 2 Hz, was used. Three undoped and three doped wafers were etched to completion at both 0 and 75 Gauss. All of the end-point diagnostic techniques mentioned in the previous paragraph were used to monitor the progress of the etch.

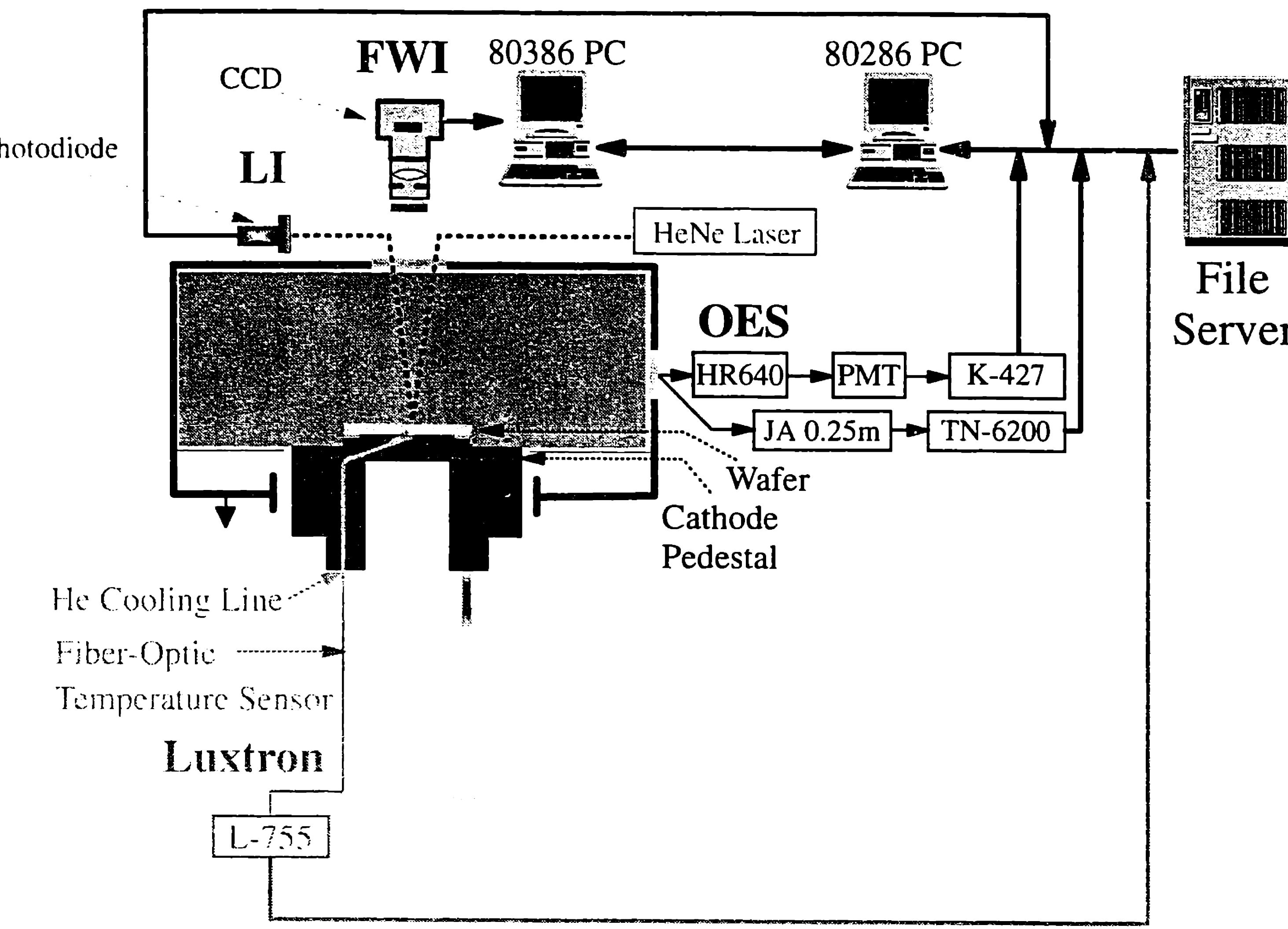


Figure 5-24: Diagnostics Installed on AME-5000

A. Full Wafer Interferometry

Both OEI and LI were performed using the full wafer interferometry technique, and the results were compared to traditional LI using a photodiode. OEI was performed with the CCD camera using an optical bandpass filter centered at 7534 Å with a FWHM of 320 Å; Cl and Br have strong emission lines at 7546 Å and 7513 Å, respectively [Reader and Corliss, 1980]. LI was performed with the CCD and with a photodiode using optical bandpass filters centered at 6328 Å. The laser was aimed close to the edge of the wafer. In order to get the laser beam into and out of the chamber at this position, a chamber lid with a 100-mm optical window was used. The area of the CCD image used for analysis was the same location as where the laser beam impinged upon the wafer surface.

As discussed in Section 5.3.1, Figure 5-7A is a plot of CCD and photodiode signals as a function of time for three cases. The black line is for OEI using the CCD. The medium line is for LI using the CCD and the light line is for LI using the photodiode. Initially, the signals are constant before the plasma is struck. At about 6 seconds, the plasma is ignited; after this, the etching of polysilicon can be seen in the periodic signal. At about 61 seconds, end-point is visible as etching of polysilicon is completed and etching of silicon dioxide begins. The difference in periodicity for the two CCD signals is due to the different wavelengths of light used for interferometry; the longer wavelength light corresponds to a longer time period in the observed signal. The same end-point time is determined from all three cases. The failure of the CCD OEI case to reach a minimum at 60 seconds comparable to the other minimums in the signal, and its

subsequent large rise, is due to an increase in optical emission intensity at end-point as will be shown later. The CCD signal using LI had a much better dynamic range than that taken using OEI. Aiming the laser at the center of the wafer and repeating the experiment showed similar results, with a different end-point time due to the difference in edge-to-center etching rate.

Interferometry using a photodiode is limited to a single spot on the wafer surface. Multiple surface locations can be analyzed using multiple photodiodes [Economu *et al.*, 1991], however, this is not practical in a manufacturing environment. Using a CCD for interferometry allows for the entire wafer surface to be viewed, and thus, the end-point time as a function of position is known. Figure 5-25 shows the end-point times across a 100-mm wafer for two diameters, located 90° apart from each other. The wafer edge clears first at about 51 seconds, while the wafer center does not clear until about eight seconds later.

As discussed in Section 5.3.6, etching with and without a magnetic field produced very different nonuniformity profiles across the wafer (Figures 5-14, 5-16). For end-point detection with FWI, when etching without a magnetic field, the start of end-point is defined as the time at which the film at the edge of the wafer begins to clear, and the end of clearing as the time when the polysilicon in the center is removed. When etching with a magnetic field, the definitions are reversed. This neglects the small patches of polysilicon near some of the clamp fingers that etches slowly. There is no difficulty in doing this because product dies would not be located near these clamp-fingers at the extreme wafer edge.

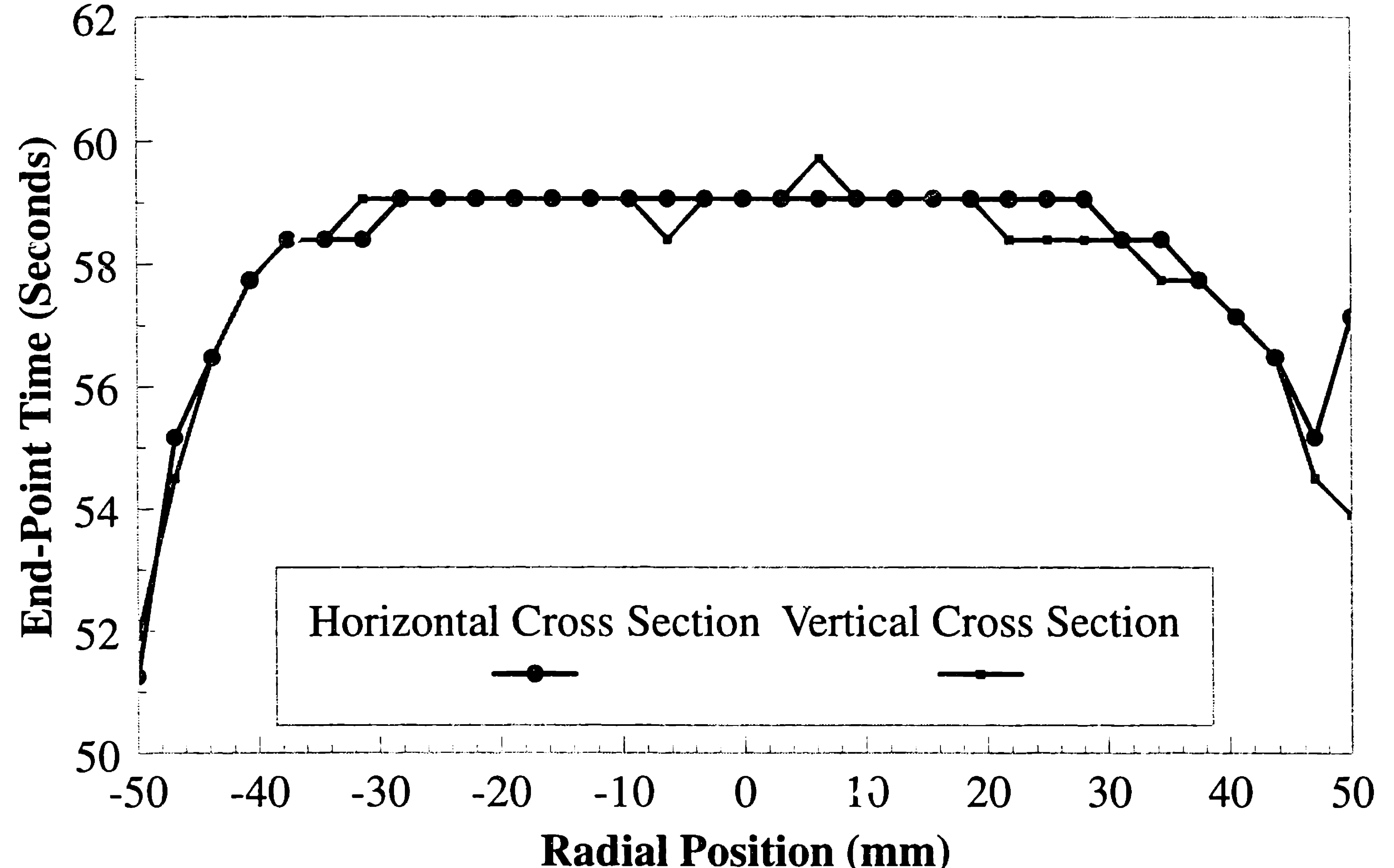


Figure 5-25: End-Point Time as a Function of Position Across a Wafer, Shown for Two Cross-Sections of the Wafer. B = 0 Gauss.

B. Photodiode Laser Interferometry

The laser beam was aimed near the center of the wafer for the remaining experiments. When etching without a magnetic field, the start of wafer clearing could not be determined using photodiode LI because, as mentioned above, the wafer cleared from edge to center. In this case, the LI signal detected the end of wafer clearing. When etching with a magnetic field, the LI signal corresponded to clearing of a large area of the wafer, not to the start or end of clearing.

C. Plasma Electrical Properties

The matching network on the AME-5000 is auto tuning. Thus, as the plasma impedance changes at end-point, the matching network tries to minimize the reflected power. This change makes end-point determination via plasma electrical parameters more difficult, since the change in the signals at end-point is small. Real power, RF voltage, RF current, plasma impedance, and RF phase angle were all tried as possible end-point monitors. Real power consistently provided the largest observable signal, with RF voltage also proving to be fairly reliable. However, the other methods sometimes did not yield a noticeable end-point signal. This was due entirely to the auto-tuning matching network. Holding the match at a fixed tuning point after the plasma had been struck yielded a better end-point signal as shown in Figure 5-26, which is a plot of plasma power as a function of time for the matching network in both auto-tune mode, and held at a fixed tuning point.

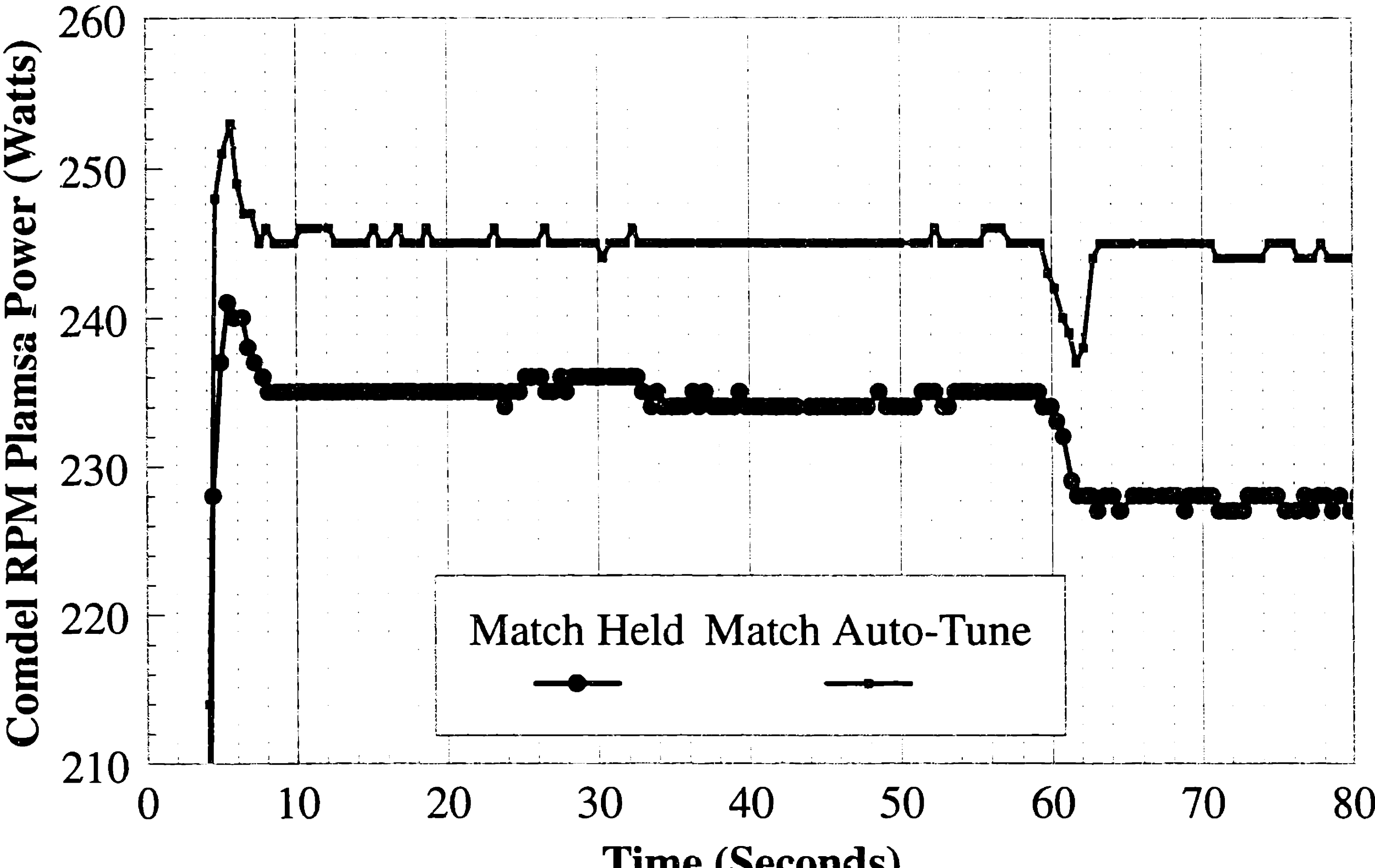


Figure 5-26: Measured Plasma Power as a function of Time, With The Matching Network in Both Auto-Tune and "Hold" Modes.

D. Optical Emission Spectroscopy

OES was done by monitoring the Cl emission line at 7547.07 Å, which is the fairly intense Cl(I) transition from $4p4s^{o}_{3/2}$ to $4s4p_{3/2}$. Figure 5-27 shows a plot of the typical behavior of the Cl emission line. There was an initial peak at plasma ignition (4 seconds) followed by a rapid drop due to consumption of the Cl reactants (8 seconds). Then, as the polysilicon film cleared, revealing the underlying silicon dioxide film (60 seconds), the Cl emission increased because Cl etches oxide significantly slower than polysilicon (the selectivity was approximately 20 with no magnetic field) and because of reduced recombination of Cl on SiO_2 [Richards, 1986]. This behavior was shown both with and without a magnetic field, as well as with other Cl emission lines.

E. Wafer Temperature

Wafer temperature monitoring during etching showed an observable end-point signal. Figure 5-28 shows a plot of the difference between wafer temperature, T_w, and cathode pedestal temperature, T_c, as a function of time both without and with a magnetic field. The wafer temperature rose about 5.5°C higher with a magnetic field than without a magnetic field; this was due to both the increased ion flux hitting the wafer and the higher etching rate. There is a non-zero difference between T_w and T_c prior to plasma ignition. This temperature difference is caused by two factors. First, radiative heat transfer from the hot walls of the etching chamber to the wafer, as evidenced by the increasing value of this temperature difference as the difference between the wall and

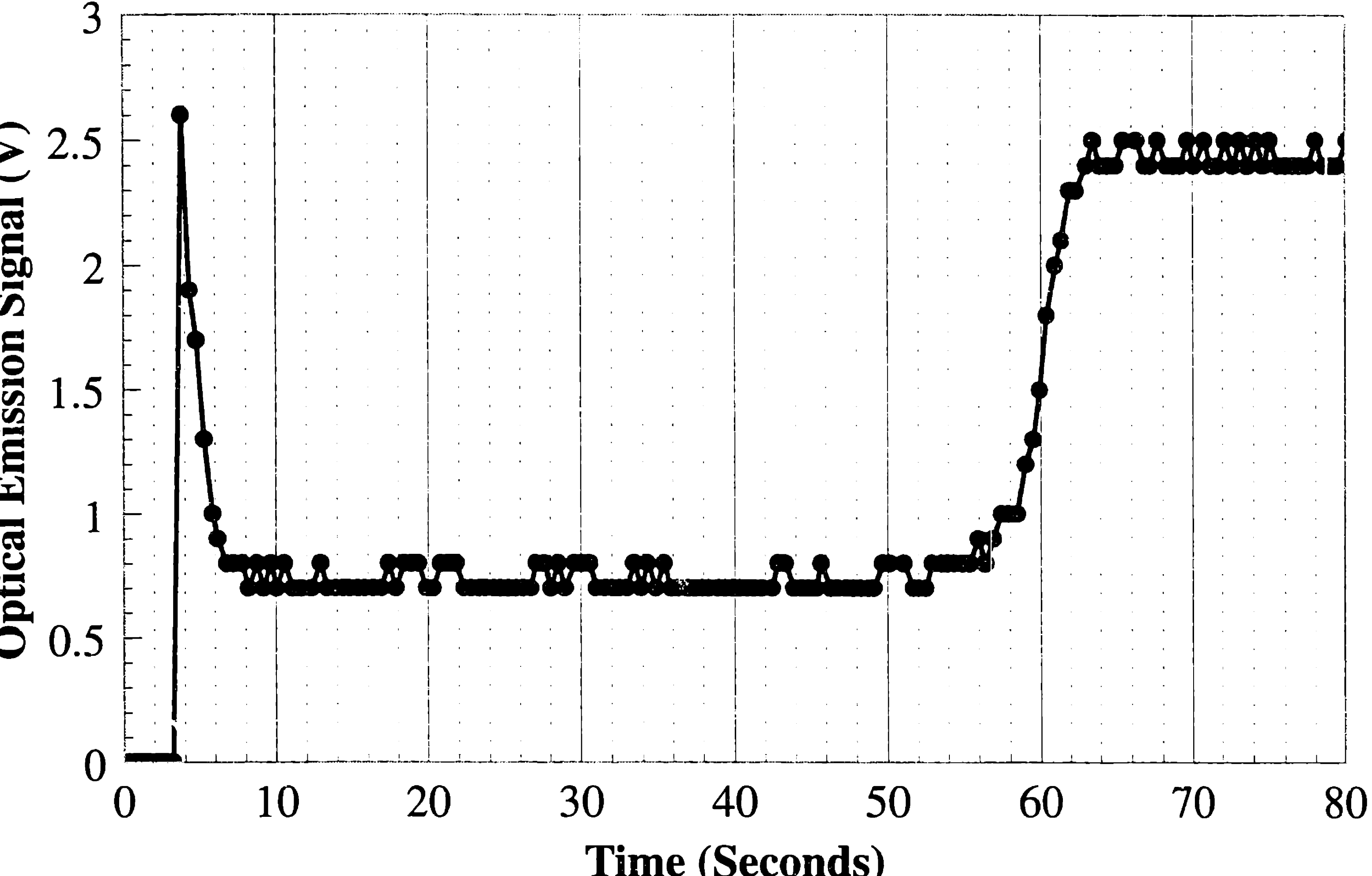


Figure 5-27: Plasma Optical Emission Signal at 7547 Å as a function of Time

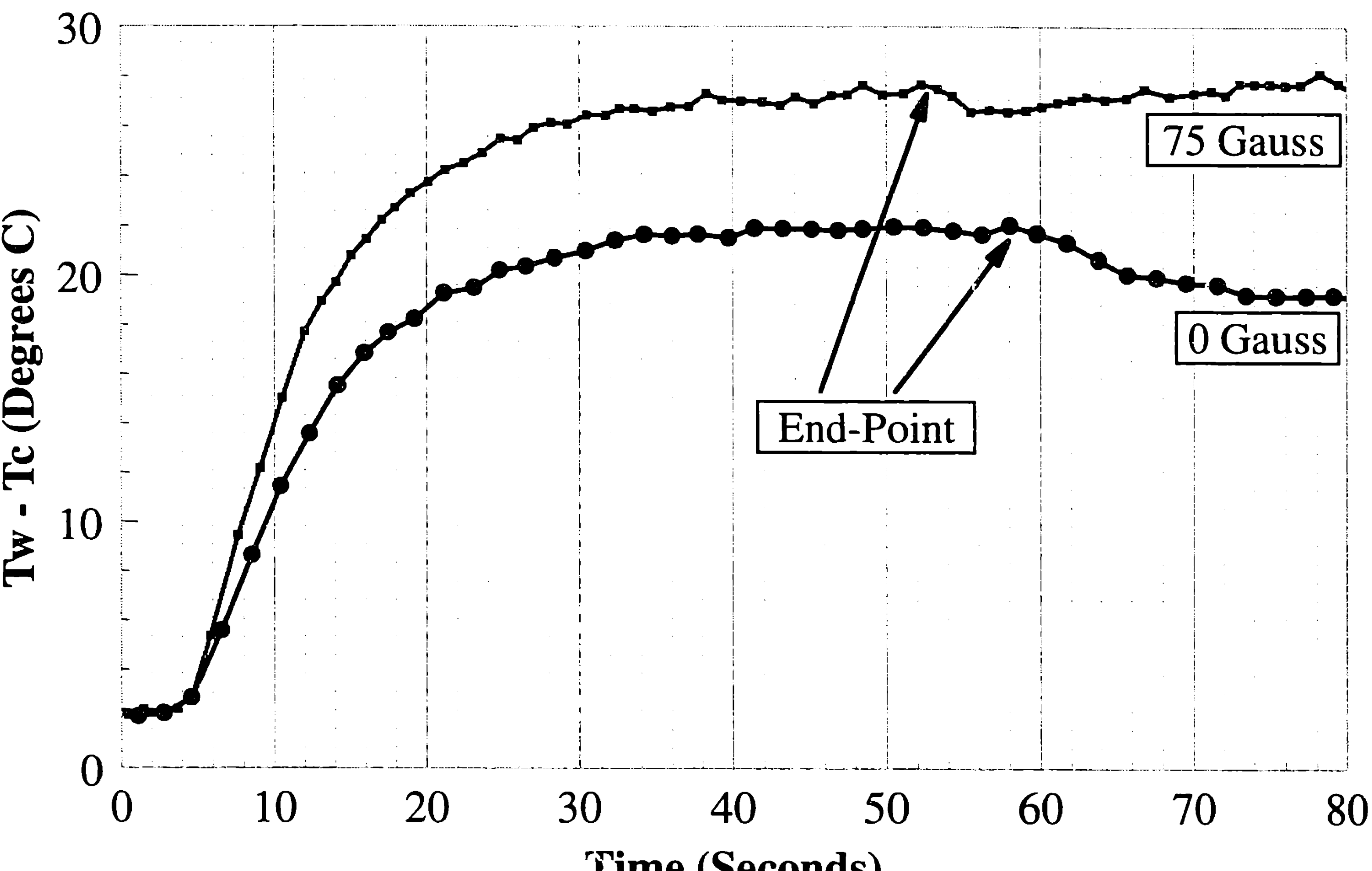


Figure 5-28 : Measured Difference Between Wafer and Cathode Pedestal Temperatures as a Function of Time both With and Without a Magnetic Field.

cathode temperatures is increased. And second, by a calibration difference in the probes.

End-point is visible at about 52 seconds with a magnetic field and about 58 seconds without one (Figure 5-28). The behavior after end-point is different for the two cases. Following end-point when etching without a magnetic field, the wafer temperature slowly decays to a new steady state value (about a 3.5 °C drop) as the oxide etching proceeds at a slow rate. The temperature drop at end-point is because silicon is no longer being volatilized at a high rate, thus the heat of reaction is no longer contributing to the wafer temperature. When etching with a magnetic field, the temperature rapidly drops by about 1 °C and then rises again. This is apparently due to the greater plasma energy flux to the wafer as indicated by the higher wafer temperature, and to the etching of the oxide which proceeds faster with a magnetic field; at a field strength of 120 Gauss, the selectivity is down to about 11.

5.4.3 Comparison of End-Point Methods

Three undoped polysilicon and three doped polysilicon wafers were etched at both 0 and 75 Gauss, with all of the above end-point techniques being used to monitor the progress of the etch. End-point times were determined after wafer processing and not in real time. For these experiments, the laser was aimed at the center of the wafer, and FWI was done using the 7534-Å bandpass filter. The start and stop of clearing for OES was determined from the 5 and 95 percent level of change in signal before and after end-point, respectively. The same algorithm was applied to wafer temperature. For plasma electrical parameters, the start and stop times of clearing were determined by the start and

stop times for signal changes. For LI, the end time was determined as the time of the final minimum in the interferometry signal. For full wafer OEI, 66 sites on the wafer were analyzed, 33 sites along a diameter, for two diameters 90 degrees apart. The end-point times were determined as a function of position, and the start and end of clearing were determined as the minimum and maximum end-point time, respectively.

The FWI technique was consistently the first one to show the start of clearing of the polysilicon film. Figure 5-29A is a plot of time to the beginning of clearing (as determined with the CCD) as a function of the end-point technique (from left to right across the graph: LI using a photodiode, Luxtron wafer temperature, OES, RF power, RF voltage, RF current, RF phase angle, plasma impedance, and FWI) for undoped polysilicon with no magnetic field. The three symbols are for each of three replicates. Some of the electrical techniques do not have three symbols shown because end-point could not be determined every time with every technique. After FWI, OES consistently was the next most sensitive technique for determining the start of clearing, about 4 seconds after the CCD. All of the others were comparable after this, indicating a start of clearing about 8 seconds after the CCD.

The time to finish clearing the wafer is shown in Figure 5-29B for undoped polysilicon with no magnetic field. FWI was the first technique to determine the end of clearing. The other techniques followed within 1 or 2 seconds, with the exception of wafer temperature which took significantly longer due to a slow decay in the temperature. Again, some of the electrical diagnostics do not yield an end-point signal for every run. Similar results were obtained for the etching of P-doped polysilicon with no magnetic field. An insert in Figure 5-29B for the Luxtron is used because of the large difference in

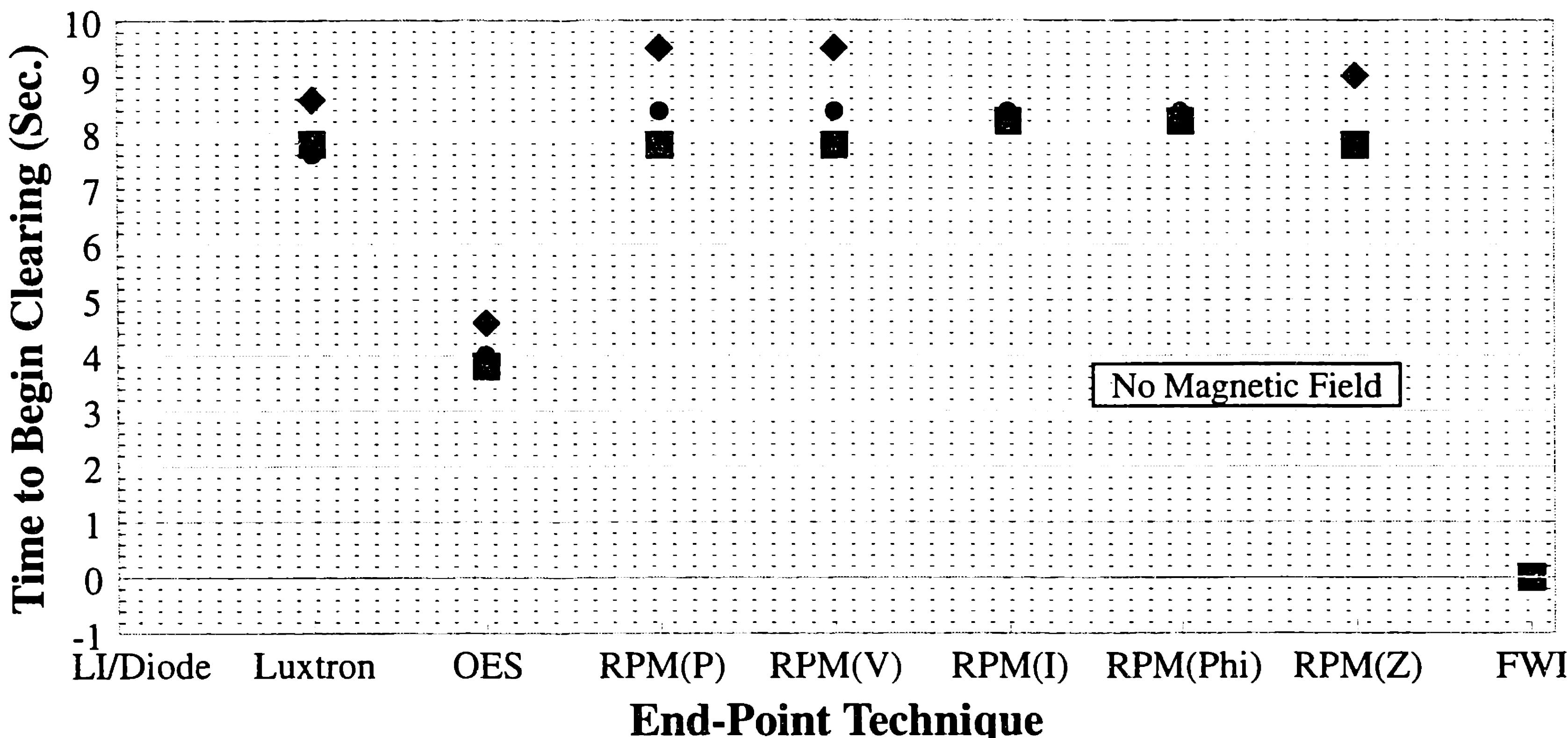


Figure 5-29A : Time to Start of Clearing for Different End-Point Methods.
B = 0 Gauss

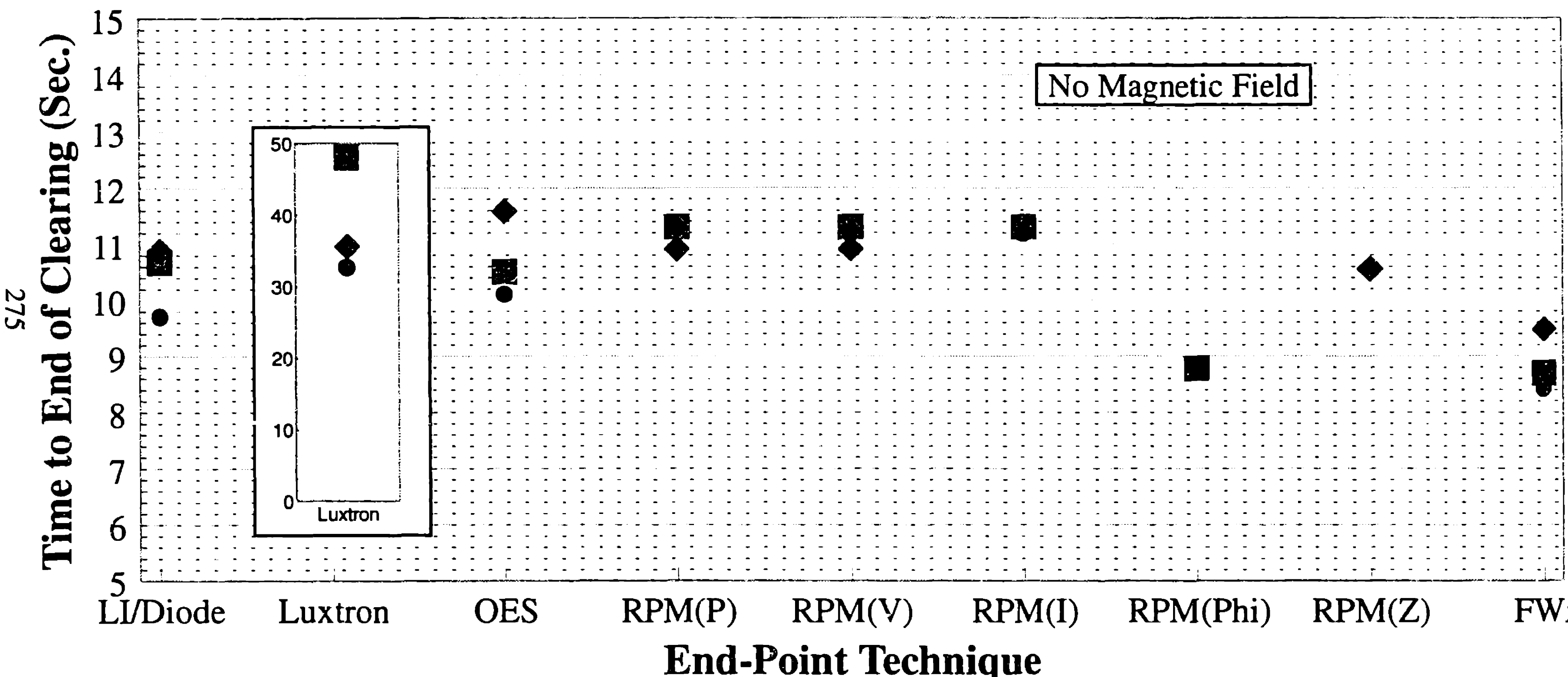


Figure 5-29B : Time to End of Clearing for Different End-Point Methods.
B = 0 Gauss

the time to finish clearing.

Etching with a magnetic field of 75 Gauss produced similar results, as is shown in Figure 5-30. The time to the start of clearing is shown in Figure 5-30A. Again, FWI consistently was the first to determine the start of clearing, with OES determining the start about 5 seconds later, followed very closely by electrical measurements, and by wafer temperature at about 9 seconds. Figure 5-30B shows the time to the end of clearing. For FWI, the end of clearing was chosen as when the wafer center clears. This neglected the time to clear the small patches of polysilicon near the clamp fingers that etched at a much slower rate than the rest of the wafer (see Figure 5-16). Using this criterion, FWI was the first to detect the end of clearing, with the other techniques following shortly thereafter. Etching of P-doped polysilicon with a 75 Gauss magnetic field showed similar results, with the time difference between the start of clearing as determined by FWI and by OES being decreased to slightly more than 2 seconds.

The times to the end of clearing as determined by OEI with the CCD and by LI with a photodiode disagree by about 1 to 2 seconds. This minor discrepancy is due to (1): the laser not being position at exactly the slowest etching region on the wafer, while the CCD can view the entire wafer; (2): differences in analysis of the data (the end-point time can vary by ±1 second depending on the algorithm used); and (3) response lag in the amplifier circuitry for the photodiode.

5.5 Conclusions

A new technique for the real-time *in situ* measurement of etching rates, etching

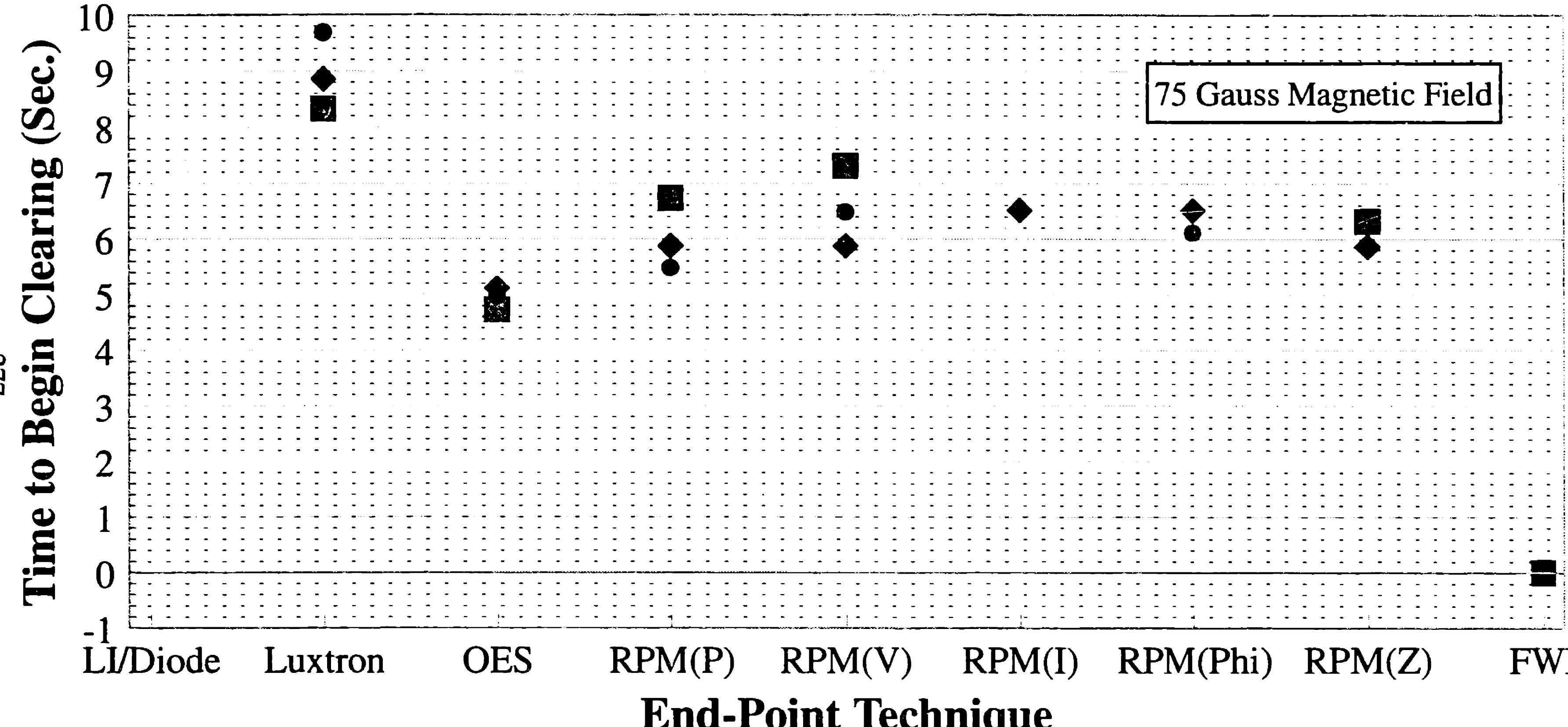


Figure 5-30A : Time to Start of Clearing for Different End-Point Methods.
B = 75 Gauss

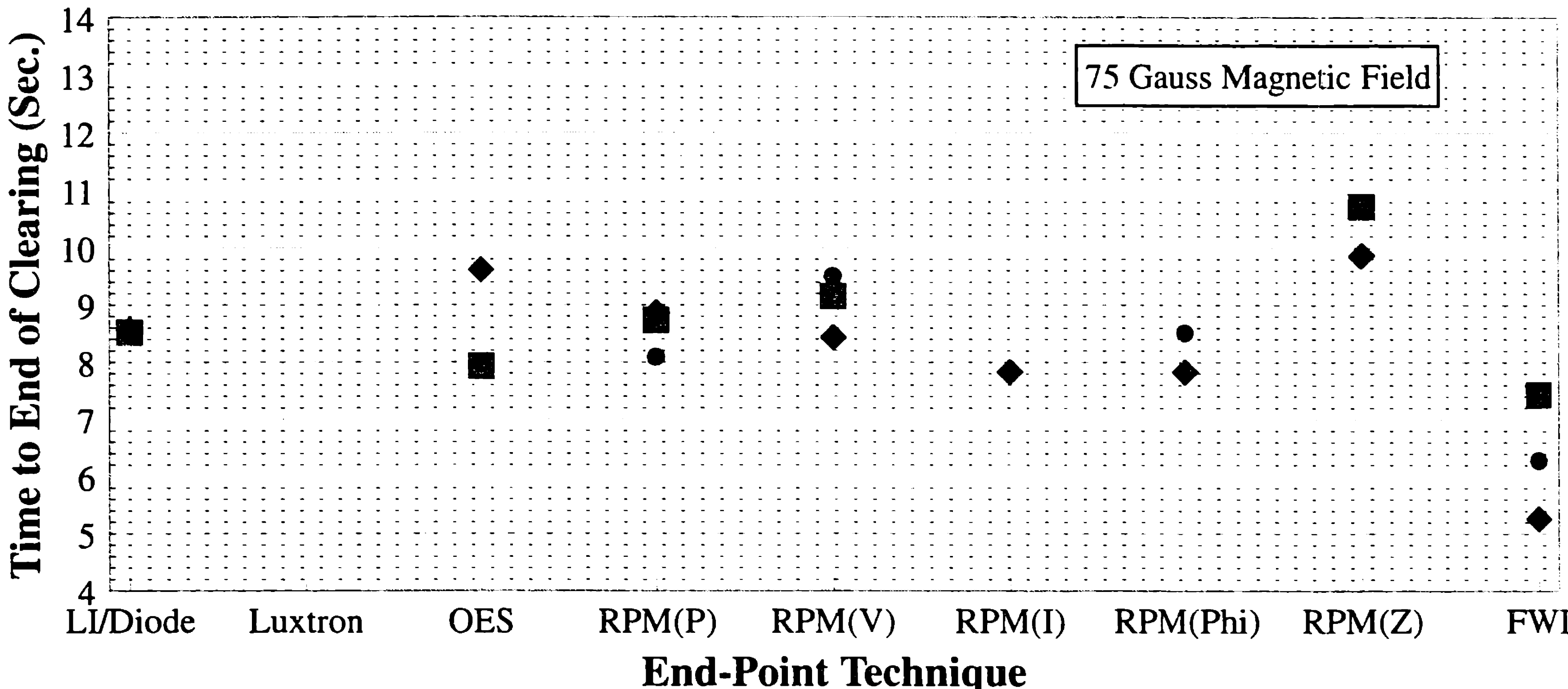


Figure 5-30B : Time to End of Clearing for Different End-Point Methods.
B = 75 Gauss

rate uniformity and selectivity using a CCD camera and plasma optical emission as the light source has been developed. The etching rates determined from this technique compare well with those measured using laser interferometry with a photodiode detector. For plasma etching, the CCD technique is applicable almost anywhere that laser interferometry is used. However, the CCD system is easier to use and much more flexible in data analysis and provides uniformity information that laser interferometry can not provide. This new technique is capable of measuring etching rates over an entire 100-mm wafer, allowing for analysis at 1 point, along a line, over a die or over an entire wafer. The technique can be easily scaled up to monitor 200-mm wafers while using a 50-mm optical view port.

The CCD technique for measuring uniformity is a valuable development tool, yielding real-time measurement of etching rate and uniformity. It has potential use as a real-time diagnostic for process control. The technique can be extended to work in other processes such as measuring deposition rates and uniformity in PECVD. Use of a laser light source can improve signal to noise, but is considerably less convenient to use, especially for large areas. This technique can also be used to determine when endpoint is reached at a number of locations across the wafer, instead of observing an average endpoint time as is done with optical emission endpoint. This technique can also be extended to measure absolute film thicknesses across a wafer in real-time *in situ* by using a tuneable interference filter to monitor a number of different wavelengths of plasma optical emission.

Several different methods of end-point determination have been compared for the etching of polysilicon on an Applied Materials Precision 5000 plasma etcher. Little

difference in the end-point times was seen between the etching of undoped and doped polysilicon, and between etching with and without a magnetic field. Some of the techniques used averaged bulk plasma or wafer properties (plasma electrical properties, optical emission spectroscopy, and wafer temperature), while others sampled wafer surface properties (laser interferometry using a photodiode and full wafer interferometry using a CCD). Only FWI yielded spatially resolved measurements.

FWI consistently determined the start of wafer clearing before any of the methods did, with OES following 4 to 5 seconds later. The other methods followed shortly thereafter, with wafer temperature having the longest delay. Similarly, FWI determined the end of clearing first, with the other methods following shortly behind.

Measurement of plasma electrical properties (actual power, RF voltage, RF current, RF phase angle, and plasma impedance) sometimes yielded an end-point signal, with power and RF voltage being the most reliable and sensitive for end-point determination. The auto-tuning matching network made end-point determination by electrical properties more difficult. Holding the matching network at a fixed tuning position after plasma ignition yielded a better end-point signal.

OES proved to be a reliable method for end-point determination. After FWI, it was the next most sensitive technique. End-point was also determined from wafer temperature. However, it consistently lagged behind the other methods due to the time required for heat-transfer. By aiming the laser for LI at the edge or the center of the wafer, only the start of clearing or the end of clearing could be determined.

Full wafer interferometry proved to be very successful for determining end-point. It has the advantage of being a spatially resolved method. Thus, the start of wafer

clearing and the end of wafer clearing can be directly determined rather than inferred from changes in the plasma bulk. Use of FWI as a real-time end-point technique is very possible and will help to control more tightly demanding plasma etching processes.

Chapter 6

Reduced Temperature Polysilicon Etching

6.1 Introduction

Wafer temperature has a significant effect on plasma etching processes. Temperature affects surface reaction rates and surface sticking coefficients, which in turn influences etching rates, etching directionality, and selectivities between films. Plasma etching typically involves a competition between spontaneous (chemical) etching, ion-assisted etching, and deposition; the relative rates of these processes can easily be altered by changing the wafer temperature.

Fluorine atoms spontaneously etch silicon at an appreciable rate. This leads to isotropic (undercut) etching of silicon in chemistries such as CF_4/O_2 and SF_6/O_2 [Flamm and Donnelly, 1981]. The standard approach to minimizing undercutting is to use a depositing chemistry that deposited a film that inhibits chemical etching on feature sidewalls; for example CF_4/H_2. The problem with this approach is that the deposited film has to be cleaned off with a separate process before subsequent process steps can be successfully done.

A different approach to minimizing isotropic etching is to reduce the isotropic etching rate relative to the ion-enhanced etching rate. This is easily done by lowering the wafer temperature because the spontaneous chemical etching rate exhibits an Arhenius-type behavior where the etching rate is exponentially dependent on temperature while the

ion-enhanced component of etching is only weakly dependent on temperature. Tachi *et al.* [1988] showed that SF_6 RIE of silicon went from being isotropic to being anisotropic at temperatures below -100°C. Similar results were seen with CF_4 and CF_3Br. The etching rate of Si in SF_6 actually increased at very low temperatures, perhaps due to an increase in the F sticking coefficient.

The temperature at which etching changes from isotropic to anisotropic is a function of the material being etched and the etchant chemistry. Tachi *et al.* [1991] reviewed low-temperature plasma etching of a number of materials in several chemistries; their results for the temperature ranges that various gases etch polysilicon anisotropically are presented in Figure 6-1. From these results, we can generalize that the more reactive an etchant is (R_{th} in Table 1-1), the lower the temperature required to stop isotropic etching.

The AME-5000 wafer cooling system standardly consists of a Neslab Coolflow chiller that has a lower temperature limit of 5°C. From Figure 6-1, we see that this might work for a pure Br_2 or HBr plasma, but would not result in anisotropic etching in a Cl_2 or F-based chemistry.

This chapter documents our modification of the AME-5000 to achieve wafer temperatures as low as -75°C. An optical temperature measuring technique was used to measure wafer temperature during wafer processing. Next I discuss the thermal analysis of the AME-5000, including heat transfer and wafer temperature histories. Finally, I discuss reduced-temperature polysilicon etching in Cl_2.

6.2 Reactor Modifications

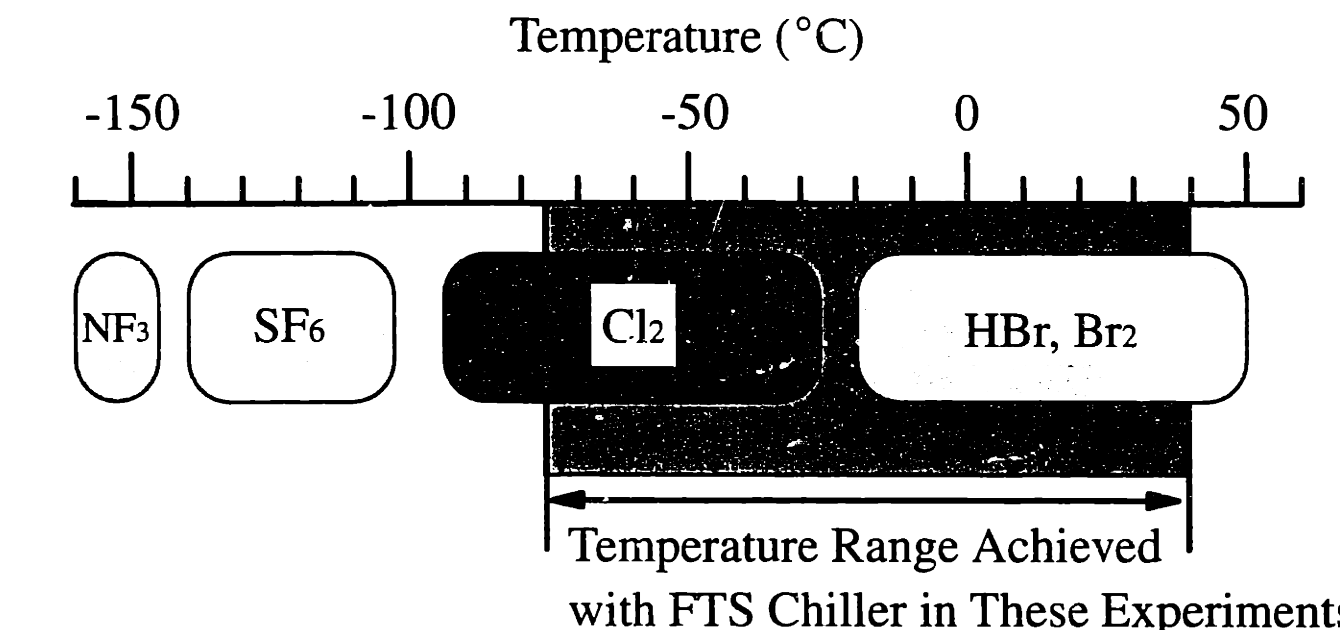


Figure 6-1: Temperature Range for Anisotropic Plasma Etching of Polysilicon. After Tachi *et al.* [1991]. Also Shown is the Cathode Temperature Range Achieved with FTS Chiller in These Experiments.

6.2.1 Temperature Control

The AME-5000 comes equipped with a recirculating chiller/heater to maintain cathode temperature. The problem with this system (Neslab Coolflow HX150, Neslab Instruments Inc. (Portsmouth, NH)) is that it is limited to the temperature range +5 to +35°C. To overcome this temperature range limitation, a different chiller/heater was added. This unit was an ultra low temperature recirculating cooler (FTS Systems Inc. (Stone Ridge, NY) #RC-210C-21) with a heater as well. It had a rated temperature range of -80 to +40 °C, with a cooling capacity of 880 Watts at -60°C, and a heating capacity of 1500 Watts. The FTS chiller was located adjacent to the polysilicon etching chamber to minimize the length of hose required to connect it to the cathode to minimize heat loss at very low temperatures.

One problem arising from reduced-temperature etching is that the AME-5000 was not designed to operate at low temperatures. The etching chamber contains a number of Viton O-rings for vacuum seals. Viton has a minimum working temperature around -40°C. During initial tests of the chiller, the plasma glow turned from blue-white at room temperature to pink (characteristic of N_2 from an air leak) at low-temperatures. The effect of temperature on the plasma glow is shown in Figure 6-2, which is a plot of the 6544.8Å optical emission from the N_2 first positive system ($B^3\Pi_g$ - $A^3\Sigma^+{}_u$ $v' = 7$, $v'' = 4$) [Pearse and Gaydon, 1976].

The solution to the air leak at low temperatures was to replace the Viton O-rings with ones made out of fluorosilicon, which remains flexible to much lower temperatures. The O-rings used were made from S-59 fluorosilicon from Advantec (Garden Grove, CA)

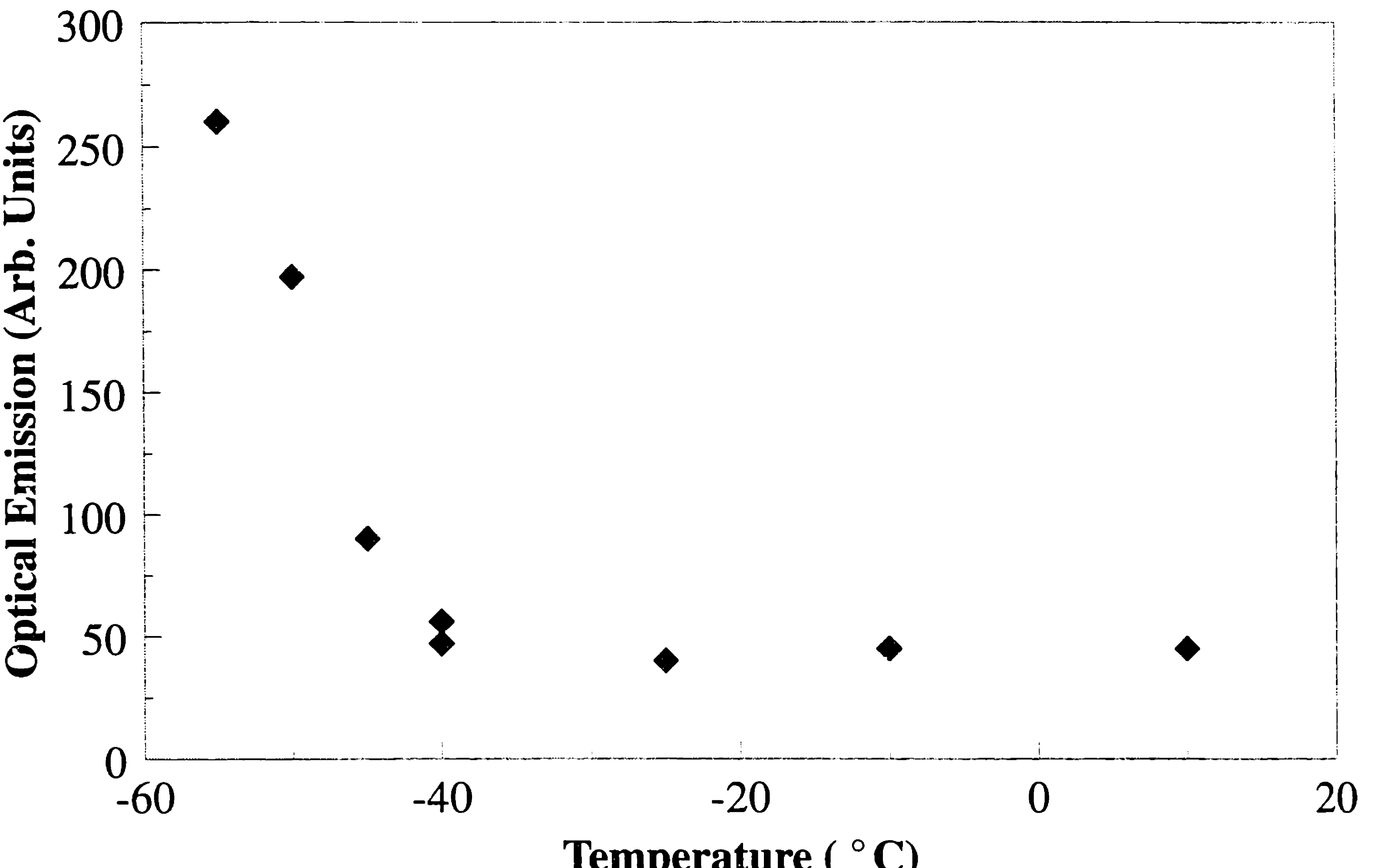


Figure 6-2: Optical Emission from N_2 First Positive System at 6544.8Å Showing Air Leak at Low Temperatures.

or from Greene Rubber (Cambridge, MA).

The "water box" enclosure below the cathode on the AME-5000 was purged with dry nitrogen to minimize the condensation of ice during low-temperature etching. Despite this, some ice still built up. The only major problem encountered during low-temperature etching was that the wafer lift assembly would often freeze in place when operating at -75°C. This made wafer transfer difficult because the cathode had to be warmed up to approximately -60°C before the lift assembly was freed up.

6.2.2 Temperature Measurement

The seemingly simple task of measuring temperature is difficult during plasma etching. The presence of RF electric fields and rotating magnetic fields make traditional temperature measuring systems such as thermocouples very difficult to implement. Measurement of wafer temperature is important because the wafer temperature rises due to energetic ion bombardment of the wafer surface and to exothermic etching reactions occurring on the wafer surface during plasma etching. The amount of temperature rise can be as small as a few degrees for low-power or well-cooled processes, and on the order of 100°C for high-power or poorly cooled processes.

An optical temperature measuring method was used to avoid the above-stated problems with temperature measurement in RF electric fields. Both the wafer temperature and the cathode pedestal temperature were measured using a Luxtron (Luxtron Corp. (Santa Clara, CA)) 755 Fluoroptic® thermometer. This technique has been proven to be effective at measuring wafer temperature during etching [Egerton,

1982]. The Luxtron 755 consists of a control unit and fiber-optic temperature probes. The probes have a temperature sensitive phosphor sensor at the tip of the fiber. The control unit provides a xenon strobe lamp to excite the phosphor and determines the temperature from the measured decay time of fluorescent radiation from the phosphor.

Recently, other non-invasive temperature measurement methods have been documented. Patel *et al.* [1991] used an infra-red (IR) television camera to monitor IR radiation in the 3-5 μm spectral range to determine wafer temperature during plasma etching. They were able to measure spatially resolved wafer temperatures during polysilicon etching in CF_4/O_2 in a homemade RF diode reactor; however, no spatially data was presented for temperature variation across a wafer.

IR interferometry at 1.152 μm was used to measure wafer temperatures in a helical resonator etcher [Donnelly et al., 1992]. Their technique is based on the change in the index of refraction with temperature changing the optical path difference in a wafer. Temperature was determined by counting interferometry fringes. The problems with this technique are that: (1) the wafer must be polished on both top and bottom, preventing the use of real wafers; (2) the observed interferometry signal is due to both changes in temperature and changes in the wafer thickness caused by etching; and (3) there is no difference between temperature increases and decreases - both produce interference fringes.

The inability to detect the direction of temperature changes occurred because only the magnitude of the interferometry signal was used. A modification of the above temperature measuring technique [Donnelly, 1993] used the phase difference in signals from a four quadrant photodiode to determine (with a high but not unity probability of

correctness) the direction of temperature change using a 1.53 μm laser. This technique still suffered from problems (1) and (2) listed in the above paragraph.

Ellipsometry has been used to measure wafer temperatures during etching of oxide films [Sampson *et al.*, 1993]. Their method relied upon the change in the index of refraction of silicon dioxide with temperature at 6328 Å. This technique worked well in very high temperature processes. However, the 20°C error was too large for use in etching processes.

The Fluoroptic® thermometry system was best suited to our needs. Its main disadvantage is that it was an invasive process, requiring that the probes touch the wafer. The other optical techniques listed above were all non-invasive.

Two MIW type probes (-190°C to +300°C operational range) were inserted into the etching chamber via the helium backside cooling line. One temperature probe touched the cathode pedestal upon which the wafer was clamped during etching, directly below the center of the wafer, and the other probe touched the center of the wafer backside (Figure 2-14).

To make the probe feedthrough, two 0.125 inch compression fittings (Swagelok SS-200-R-2) were welded onto the He feedthrough line (Applied Materials Part # 0040-00543) at an angle of 40° off the axis of the feedthrough (Figure 6-3). A 0.75 inch long section of 0.125 inch O.D. Teflon® rod was center drilled using a #55 drill (0.0520 inch). The probes were inserted through the Teflon® sleeves and through the compression fittings. The vacuum seal was made by using a stainless steel ferrule to clamp down on the Teflon® sleeves. The seal made this worked quite well; for a chamber leak-up test, over 12 minutes, no pressure rise was observed (+0.1 mTorr sensitivity). This

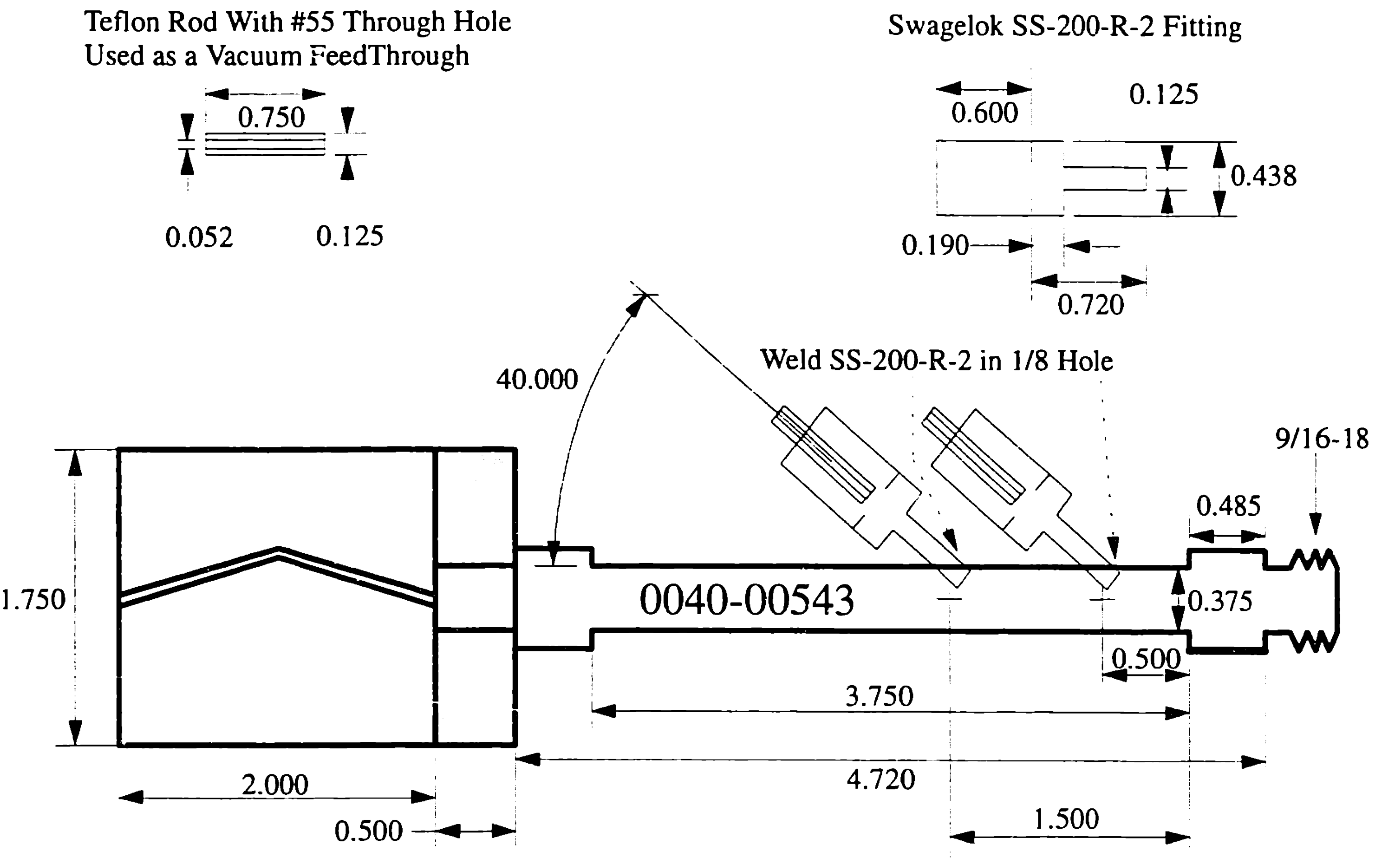


Figure 6-3: Temperature Probe Feedthrough for AME-5000.

corresponded to a chamber leak rate of less than 1.8 x 10^{-6} Torr-liter/sec (1.4 x 10^{-4} SCCM).

Alternatively, a probe was placed on the RF feedthrough Cu post on the back of the cathode. However, the temperature here was very stable due to the large cooled thermal mass between the plasma and the point of measurement.

In order to assure good thermal contact between the wafer and the cooled cathode pedestal, the wafer was clamped onto the pedestal. The gap between the wafer and the pedestal was filled with up to 10 Torr of He. The He was contained by an O-ring lip seal at the edge of the wafer .

6.3 Thermal Analysis of AME-5000

During plasma etching, the wafer temperature rises due to energetic ion bombardment; the heat flux to the wafer from the plasma, q_p, is given by

$$q_p = J_i \langle E_i \rangle \quad \textbf{(6-1)}$$

where J_i is the ion flux (current density) (Amp/cm^2) and $\langle E_i \rangle$ is the average ion energy (Volts). q_p is a fraction of the power input to the plasma (measured at the generator or at the cathode).

There are three possible mechanisms for heat transfer from the wafer. They are: (1) radiation; (2) convection to the feed gas flowing across the wafer front; and (3) conduction across the wafer backside to a cooled pedestal. A drawing showing the cathode assembly was shown earlier (Figure 2-14). These three phenomena are

investigated below.

6.3.1 Radiative Heat Transfer

Heat transfer in vacuum can occur via an electromagnetic mechanism known as radiation. The heat flux from the wafer due to radiation, q_r, is given by

$$q_r = e(T)\ \sigma\ (T^4 - T_0^4) \tag{6-2}$$

where *e(T)* is the emissivity evaluated at wafer temperature T, T_0 is the chamber temperature, and σ is the Stefan-Boltzman constant (5.669 x 10^{-12} Watts cm^{-2} K^{-4}) [Bird *et al.*, 1960]. Typical values of *e(T)* are in the range of 0.2 to 0.4 [Visser, 1989]. An estimate of the maximum radiative heat flux is calculated using T_0 = 40°C, T = 100°C, and *e(T)* = 0.4 yields $q_{r,max}$ = 0.022 Watts cm^{-2}. For a wafer area of 80 cm^2, this corresponds to a maximum heat transfer rate of 1.8 Watts. For smaller temperature differences, or lower emissivities, this contribution will be significantly smaller.

A small temperature difference was observed between the two temperature probes installed in the AME-5000. This difference may have originated from: (1) different length probes being used (one was 10-meter and one was 2-meter); or (2) crimping of the probe caused by the vacuum feedthrough. This temperature difference was affected by the difference in chamber wall temperature and cathode temperature (Figure 6-4). A difference of approximately 1.6°C between the wafer and pedestal temperatures was obtained when the chamber wall and cathode temperatures were equal. The temperature difference increased with increasing difference in the wall and cathode temperatures. A

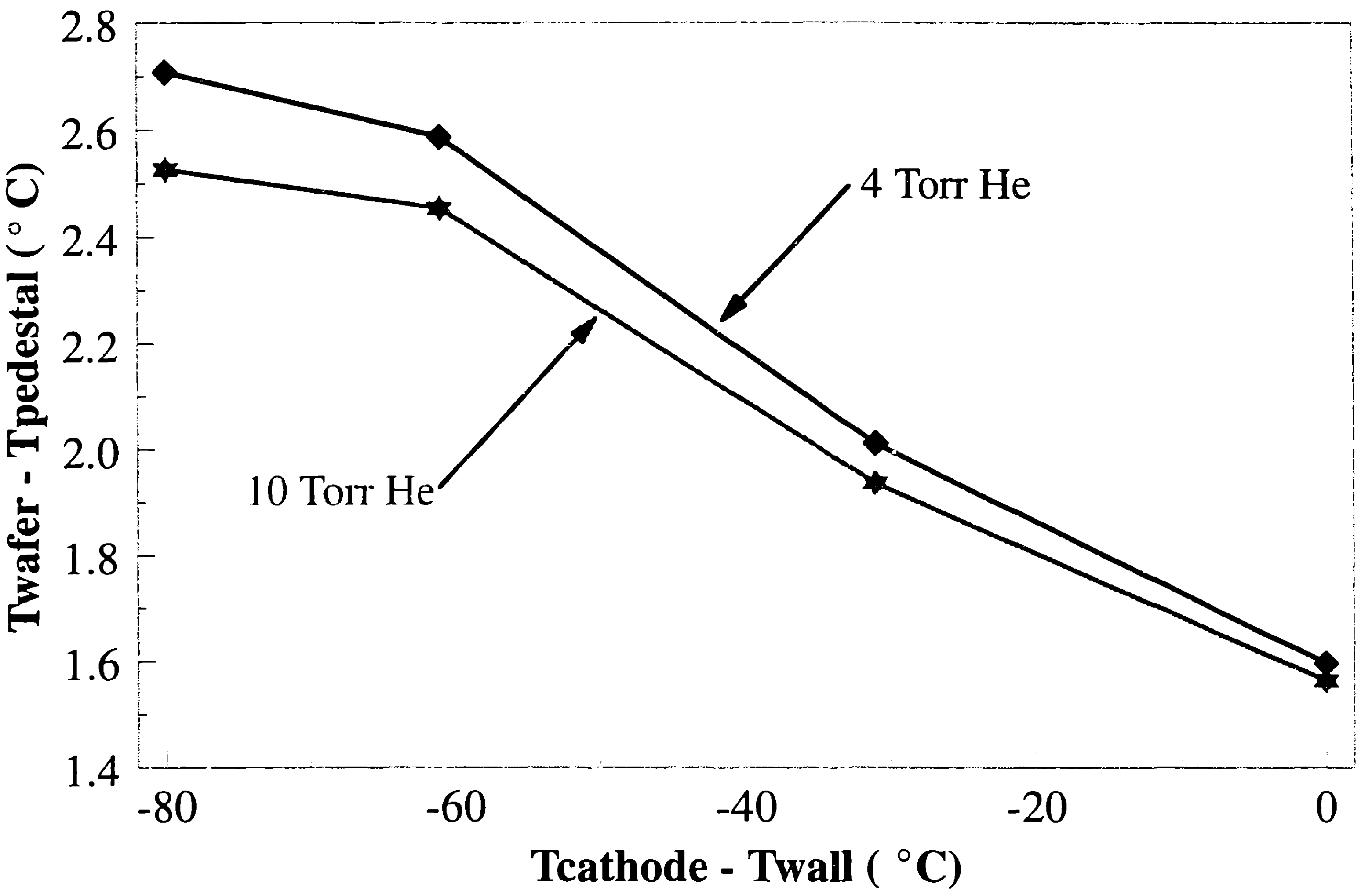


Figure 6-4: Effect of Radiative Heat Transfer and Wafer Backside Conduction on Temperature Difference Between Two Luxtron MIW-type Temperature Probes.

maximum difference on the order of 2.5°C was obtained for a wall-cathode temperature difference of -80°C.

6.3.2 Convective Heat Transfer to Feed Gas

A second mechanism for heat transfer from the wafer is by convection to the flowing feed gas, assuming that the neutral temperature of the gas is less than the wafer temperature. During etching, the neutral temperature may be on the order of 400-500K, which is above the wafer temperature. However, after the plasma is extinguished, the neutral gas is at room temperature.

The effect of gas pressure on wafer cooling is shown in Figure 6-5, which is a plot of the wafer temperature after plasma extinction as a function of the chamber pressure. The wafer temperature is only slightly sensitive to the chamber pressure, with higher chamber pressures leading to better heat transfer, and thus lower wafer temperatures. No difference is seen between chamber pressures of 13 and 50 mTorr. From this information, we conclude that convective heat transfer to the feed gas is a minor component of heat transfer from the wafer; in other systems, this component may be important [Visser, 1989].

6.3.3 Convective Heat Transfer To Pedestal - Conduction Across a Static Pressurized Gap

The third and dominant mechanism for heat transfer from the wafer in the

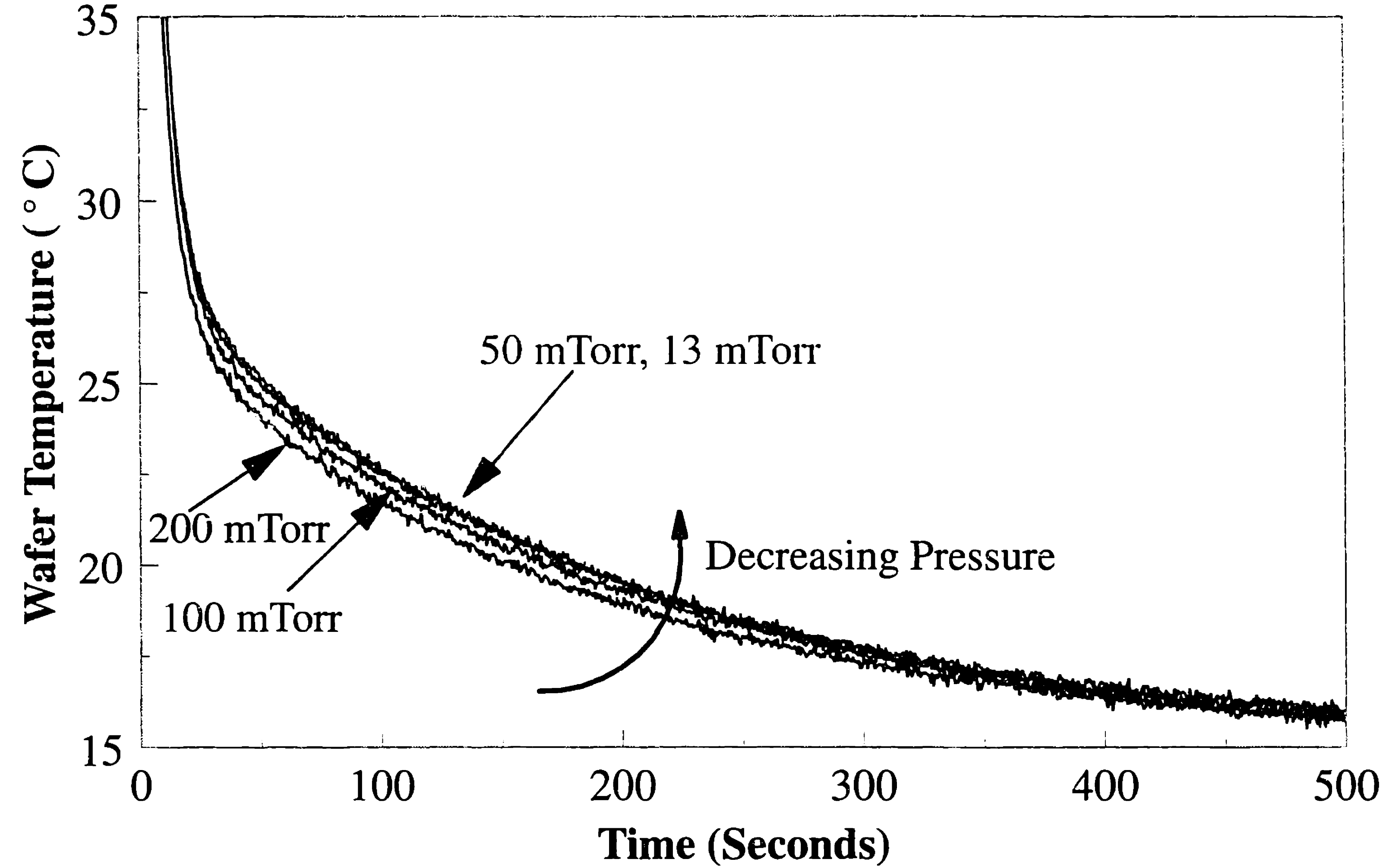


Figure 6-5: Effect of Chamber Pressure on Wafer Cooling After Plasma Etching. 75 SCCM SF_6, 4 Torr He.

AME-5000 is by conduction across the wafer backside, through relatively high pressure He to the pedestal which is cooled. Since the heat transfer occurs across a pressurized gap and not by direct contact, Fick's law is not the best representation for the heat flux. Instead, the heat flux by conduction, q_c is given by

$$q = (T - T_p) \qquad (6\text{-}3)$$

where T_w is the wafer temperature, T_p is the pedestal (nominally the chiller temperature), and h is the heat transfer coefficient. Multiplying this by the heat transfer area, A, yields the rate of heat transfer.

The functionality of h depends on the Knudsen Number, N_{Kn}, which is given by

$$= \frac{\lambda_{He}}{g_0} \qquad (6\text{-}4)$$

where λ_{He} is the mean free path of He atoms in the pressurized gap, and g_0 is the gap size. For $N_{Kn} << 1$ (continuum regime) there are a significant number of gas phase collisions. In this case, h is given by [Smith, 1984]

$$h \sim \frac{N \lambda_{He}}{g_0 \sqrt{M}} \qquad (6\text{-}5)$$

where N is the gas density, and M is the molecular weight of He. The product $N\lambda_{He}$ is independent of pressure. Thus, h is independent of pressure, and inversely proportional to gap spacing. For $N_{Kn} >> 1$ (rarefied regime) there are essentially no gas phase collisions. In this case, h is given by [Smith, 1984]

$$h \sim N c_V \bar{v} \gamma \sim \frac{N \gamma}{\sqrt{M}} \quad \textbf{(6-6)}$$

where c_V is the constant volume heat capacity, $\bar{v}$ is the mean speed, and γ is the accommodation coefficient. Thus, *h* is proportional to the gas density (and thus to the pressure), and independent of gap spacing. The range of $0.1 < N_{Kn} < 10.0$ is a transition regime between the two extreme behaviors. Theoretical values of *h* for He in a plasma etcher have been calculated by others [Wright et al., 1992].

The mean free path for He at 300K as a function of pressure (0.1 - 20 Torr) is plotted in Figure 6-6. Also shown is the calculated wafer deflection (Section 6.3.5) which is approximately equal to g_0. From these, N_{Kn} is calculated. The pressure range of interest on the AME-5000 is from 1 to 10 Torr He. This corresponds to N_{Kn} between 0.03 and 3.0. Thus, heat transfer is in the transition regime for most of the range of interest.

6.3.4 Heat Transfer Coefficient Determination

From the discussion in the previous section, we see that *h* is the important parameter determining heat transfer on the AME-5000. The heat flux to the wafer surface faces two possible resistances to heat transfer. The first is across the wafer, and the second is across the wafer backside gap. The relative importance of these two resistances is given by the Biot number, Bi, which is the ratio of the resistance of the solid (wafer) to the resistance of the fluid (He), and is given by [Deen and Brown, 1988],[Fahien, 1983]

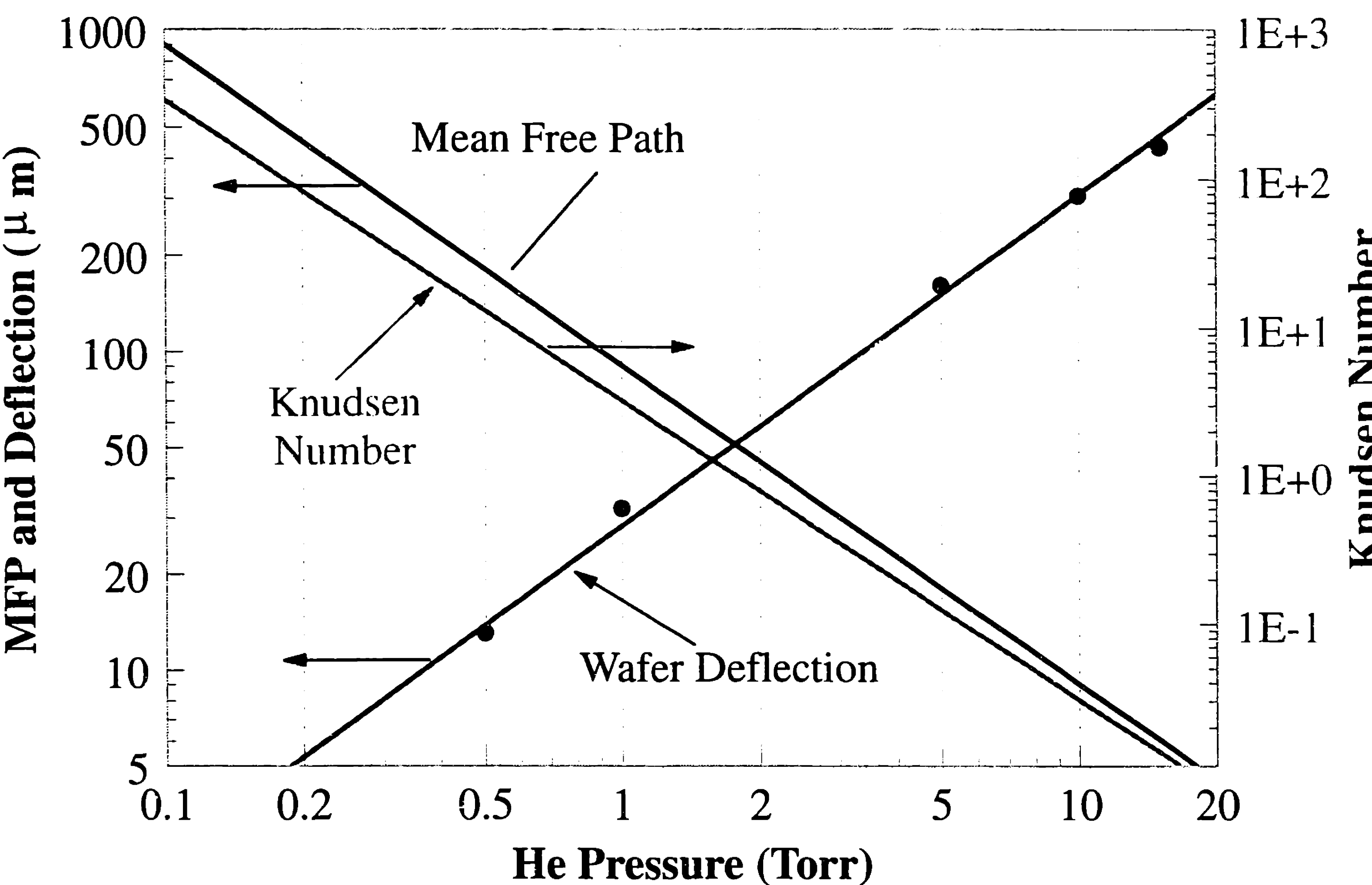


Figure 6-6: Calculated Mean Free Path of He (λ_{He}), Wafer Deflection (g_0), and Knudsen Number (N_{Kn}) as a Function of He Backside Pressure.

$$\mathrm{Bi} = \frac{h\, t_w}{k_{Si}} \tag{6-7}$$

where t_w is the wafer thickness and k_{Si} is the thermal conductivity of silicon. For Bi >> 1, the resistance of the solid dominates, and the fluid is isothermal. For Bi << 1, the resistance of the fluid dominates and the solid is isothermal. For .1 < Bi < 1, both resistances are important. For 100-mm wafers, t_w = 0.05 cm. Using h = 0.01 Watts/cm^2-K (calculated below), and k_{Si} = 1.68 Watts/cm-K, the calculated value of Bi is 3.0 x 10^{-4}. Thus, Bi << 1 so that the wafer can be treated as isothermal across the wafer thickness; The temperature measured at the back of the wafer is identical to that at the front.

A transient heat balance on the wafer after plasma extinction yields

$$(t_w A)\,\rho\, c_p \frac{dT_w}{dt} = -hA(T_w - T_p) \tag{6-8}$$

where ρ is the density of Si (2.33 g cm^{-3}), c_p is the constant pressure heat capacity (0.168 cal g^{-1} K^{-1}), t is time, and the other terms are as defined above. Solving for $T_w(t)$ by separation of variables and integration, with the initial condition $T_w = T_{w0}$ at t = 0, yields

$$T_w = T_p + (T_{w0} - T_p)\exp\left[\frac{-h}{\rho\, c_p t_w}\, t\right] \tag{6-9}$$

This suggests one method to measure the heat transfer coefficient. Measure the wafer temperature as a function of time during cool down after plasma extinction, or drop a room temperature wafer onto a chilled cathode. An exponential fit transient response of the wafer temperature would contain h which is calculated as

$$h = \rho\, c_p\, t_w\, \beta \tag{6-10}$$

where β is from the exponential fit of the form $\exp(-\beta t)$.

The heat transfer coefficient was determined using the method of dropping the wafer onto a chilled cathode and measuring the temperature as a function of time. A sample of the measured responses is shown is Figure 6-7. Plotted in this figure are the wafer (T_w) and pedestal temperatures (T_p), and their difference, (T_w - T_p). Also shown is the exponential fit to the data used to calculated the heat transfer coefficient. The exponential fit to both T_w and (T_w - T_p) yielded similar values of *h*.

The process detailed above was repeated for He pressures of 1 to 10 Torr. The heat transfer coefficients as a function of the He pressure are plotted in Figure 6-8. The circles represent calculation of *h* by dropping the wafer onto the chilled cathode ("cold" calculation), while the squares represent calculation of *h* by measuring temperature fall after plasma extinction ("hot" calculation). No difference is seen in the two methods. The data have been fit to the following equation

$$h = 35.6\, p^{0.502}\ [Watts\ m^{-2}\, K^{-1}] \tag{6-11}$$

where *p* is the He pressure in Torr. The exponent for the pressure dependance indicates that the data are in the transition regime, being neither proportional to, or independent of pressure.

Wright *et al.* [1992] presented an analysis of a newly designed low temperature etch chuck for 150-mm diameter wafers. Using an equation identical to Equation 6-9; they measured *h* using both the "cold" and "hot" calculation methods. Their results are

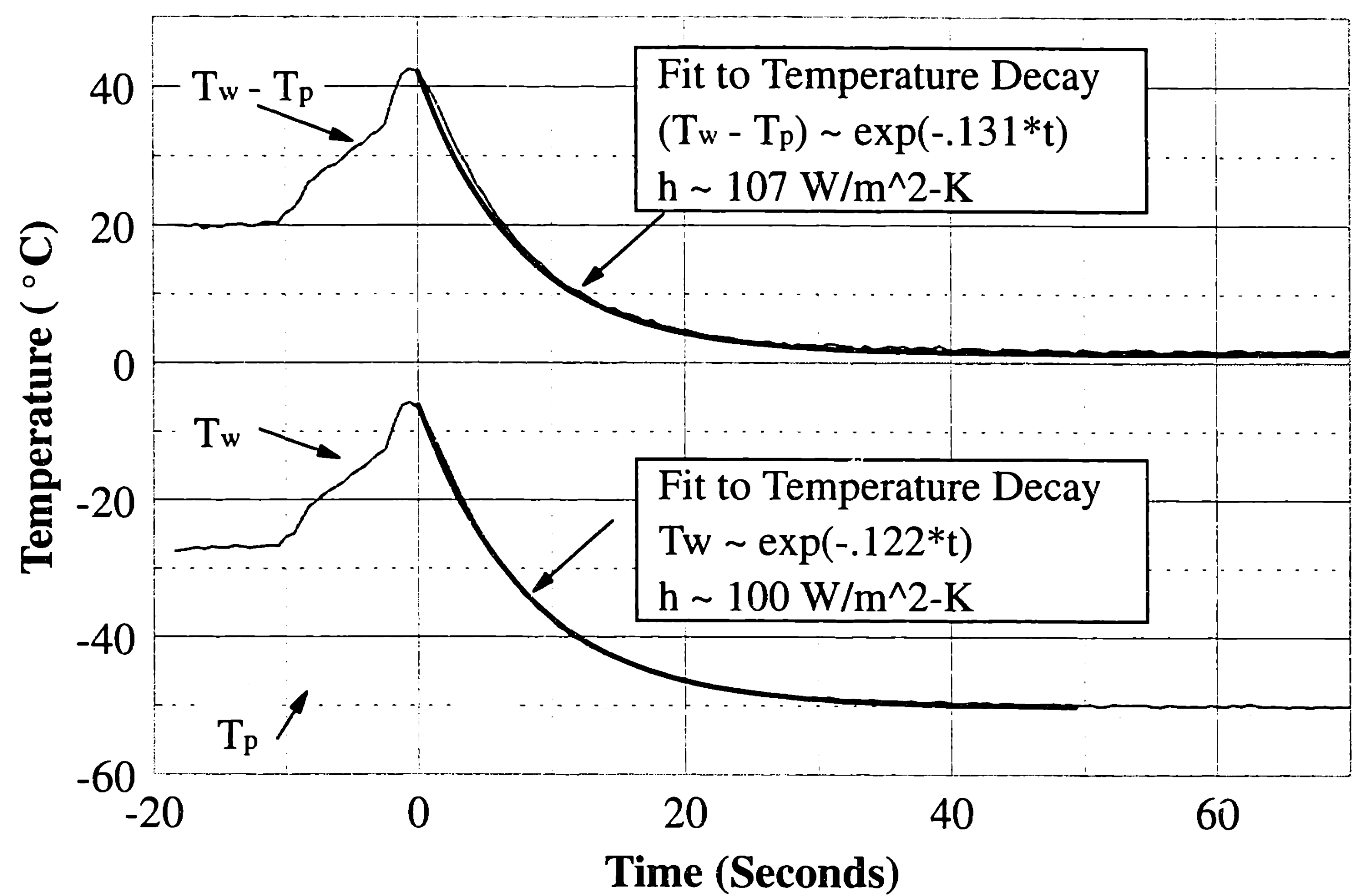


Figure 6-7: Typical Temperature Response for HTC Measurement. Room Temperature Wafer Placed on Cold Cathode, then He Turned on. Also Shown is Fit to Data.

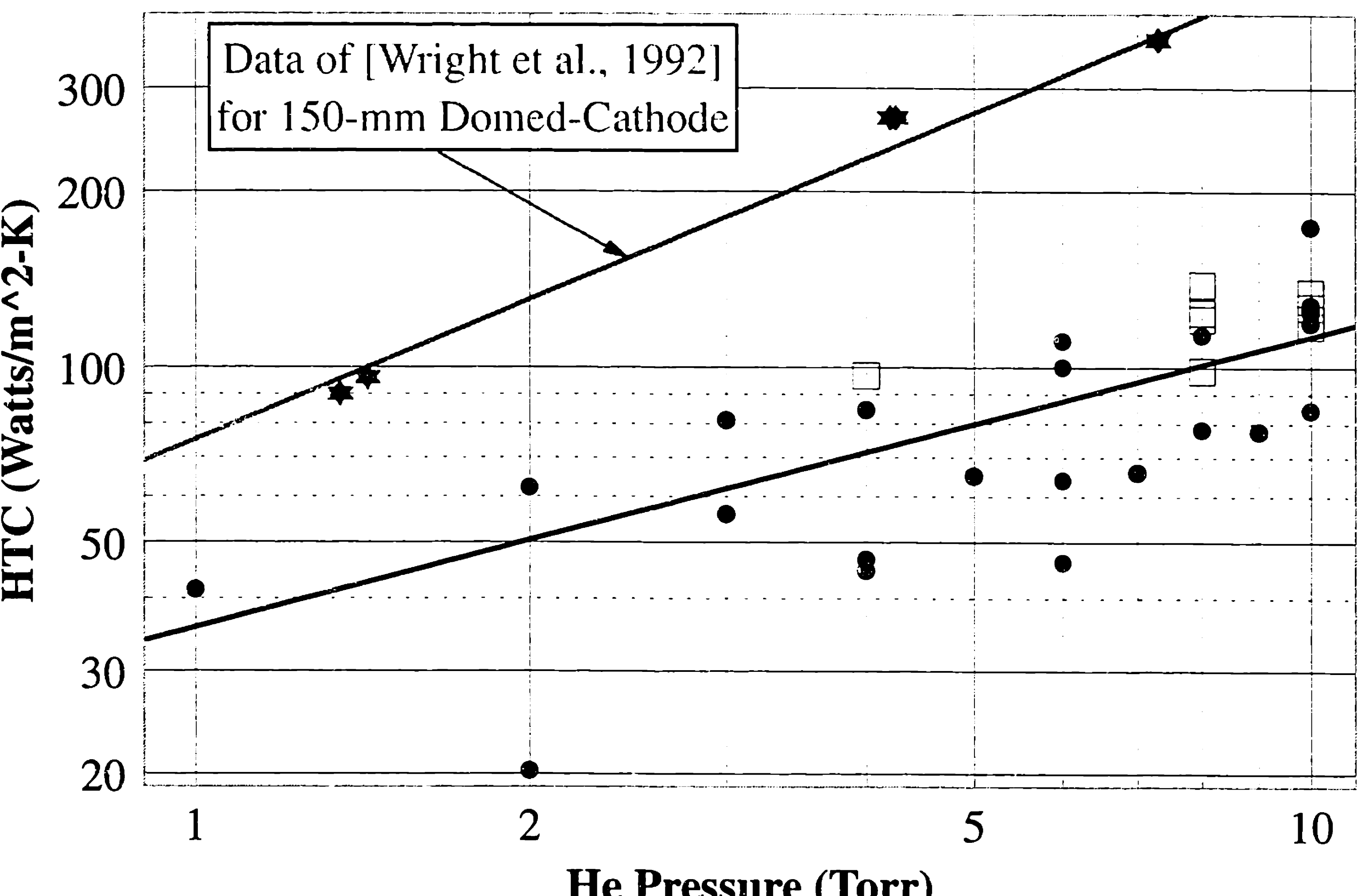


Figure 6-8: Wafer Heat Transfer Coefficient as a Function of He Backside Pressure. Circles are "Cold" Calculations, and Squares are "Hot" Calculation.

also shown on Figure 6-8. For their system, the data were fit to

$$h = 75.0\, p^{0.80} \; [Watts\; m^{-2}\, K^{-1}] \qquad \textbf{(6-12)}$$

In this case, the data were closer to linear in pressure, indicating that they were more towards the rarefied regime that our results. The difference in their data and ours arises because of the cathode design. The cathode pedestal on the AME-5000 is flat. Because of this, a significant gap exists between the pedestal and the wafer (Section 6.3.5). The pedestal of Wright *et al.* [1992] was domed; it was curved so that the gap is minimized. This smaller gap shows up as: (1) an increase in N_{Kn}; and (2) an increase in h - at 10 Torr the difference is almost a factor of 4. In the absence of He backside cooling, others have measured heat transfer coefficients on the order of 10 Watts m^{-2} K^{-1}, which increased to the order of 100 Watts m^{-2} K^{-1} when the wafer backside was coated with vacuum oil [Patel *et al.*, 1991]. Others have observed essentially no heat transfer through the wafer backside in the absence of He [Visser, 1989].

6.3.5 Wafer Deflection

When a wafer is clamped onto the cathode pedestal, and He backside cooling is applied, the He pressure exerts a force on the wafer which is significant enough to cause deflection of the wafer. The deflection distance is on the order of 100 μm at 10 Torr for a 100-mm wafer. This distance, though small, can have a significant effect on the heat transfer coefficient. For this reason, the wafer deflection was quantified.

Formulas for the deflection of a circular plate under uniform loading have been

developed by others [Roark and Young, 1975] (this reference is the source for all deflection equations presented here). These deflection formulas assume a form of support at the edge of the plate. Both "simple support" and "fixed" models were calculated to assess the difference in the two techniques. An average value from these two calculations was used as the wafer deflection. The deflection equations also depend upon the amount of deflection. Therefore, the deflection amount was calculated using both small and large deflection equations to compare the difference. For low He pressures (less than 1 Torr), the small deflection equations applied; for larger He pressures, the large deflection equations applied.

For small deflections, with a uniformly distributed load, and simple support, the maximum wafer deflection, y_c, is given by

$$y_c = \frac{q a^4 (5 + \nu)}{64 D (1 + \nu)} \qquad \textbf{(6-13)}$$

where q is the uniform load (the He pressure, lb in^{-2}), a is the wafer radius, ν is Poisson's ratio (0.42 [Sze, 1983]), and D (the plate constant) is given by

$$D = \frac{E t_w^3}{12(1 - \nu^2)} \qquad \textbf{(6-14)}$$

where E is Young's modulus (10890 kg/mm^2 [Sze, 1983]). For fixed support, y_c is given by

$$y_c = \frac{q a^4}{64 D} \qquad \textbf{(6-15)}$$

For the large deflection model, the deflection is determined from the following equation

$$\frac{qa^4}{Et_w^4} = K_1 \frac{y_c}{t_w} + K_2\left(\frac{y_c}{t_w}\right)^2 \tag{6-16}$$

where K_1 and K_2 depend upon the type of support. For simple support, they are

$$K_1 = \frac{1.016}{1 - \nu} \qquad K_2 = 0.376 \tag{6-17}$$

and for fixed support they are given by

$$K_1 = \frac{5.33}{1 - \nu^2} \qquad K_2 = \frac{2.6}{1 - \nu^2} \tag{6-18}$$

The wafer deflections calculated for a 0.05-cm thick, 100-mm diameter Si wafer are shown in Figure 6-6. At 1 Torr He, the deflection is about 30 μm, and at 10 Torr He, it is about 300 μm. The wafer deflection was found to fit the following form

$$y_c = 28.3 p^{1.04} \ [\mu m] \tag{6-19}$$

where p is the He pressure in Torr.

The deflection becomes more severe as the wafer size increases. A 0.675-cm thick, 150-mm diameter wafer experiences a deflection of about 600 μm at 10 Torr He, and a 0.75-cm thick, 200-mm diameter wafer experiences a deflection of 1100 μm at 10 Torr He.

These large deflections point to the need for domed cathodes in single wafer

etchers for large wafer diameters. Without accounting for the wafer deformation, the heat transfer between the wafer and cathode will decrease, possibly to unacceptable levels. Other possibilities are using what is known as an "electrostatic chuck" that electrically clamps a wafer to the cathode pedestal. This uniform clamping helps decrease deflection (unlike traditionally edge clamping).

6.3.6 Radial Temperature Variations

The wafer temperature may vary radially because of (1) changes and the gap spacing (and thus h) with radial position, and (2) because of heat transport. The second cause was examined to determine if the measured wafer temperature could differ greatly from the extreme wafer temperatures.

A steady state heat balance on the wafer during etching assuming azimuthal symmetry and Bi << 1 (T = T(r) only) yields

$$\frac{1}{r}\frac{d}{dr}\left(r\frac{dT}{dr}\right) = -\frac{\Phi_H}{k_{Si}} \tag{6-20}$$

where k_{Si} is the thermal conductivity of Si, and Φ_H is the heat added to the wafer per unit volume due to the plasma. Φ_H can be expressed as the total heat input to the wafer divided by the wafer volume ($\pi a^2 t_w$). This equation can be solved subject to the boundary conditions

$$T|_{r=a} = T_0$$
$$\left.\frac{dT}{dR}\right|_{r=0} = 0 \tag{6-21}$$

Which yields a radial temperature profile of

$$T(r) = T_0 + \frac{\Phi_H}{4k_{Si}}(a^2 - r^2) \tag{6-22}$$

The maximum temperature difference, ΔT_{Max}, between r = 0 and r=a is equal to

$$\Delta T_{Max} = \frac{\Phi_H a^2}{4k_{Si}} = \frac{P_{in}}{4\pi t_w k_{Si}} \tag{6-23}$$

where P_{in} is the power input to the wafer. This equation indicates that ΔT_{Max} is a function of the wafer thickness, wafer thermal conductivity, and the amount of power being input into the wafer. Calculated values of ΔT_{Max} for several input power inputs are listed in Table 6-1.

Table 6-1

Maximum Radial Wafer Temperature Differences

P_{in} (Watts)	ΔT_{Max} (°C)
10	6.7
25	16.9
200	134

From the data of Table 6-1, it is clear that ΔT_{Max} may be significant, if the power input to the wafer is significant. There is no way to modify this except by changing the wafer thickness. But, as will be shown in Section 6.3.8, the actual power input is small, on the order of 10 Watts so that the radial temperature variations are expected to be less than 10°C.

6.3.7 Wafer Temperature During Plasma Etching

The two parameters of interest when observing wafer temperature during plasma etching are the maximum temperature a wafer reaches during processing, and the rate at which it reaches that temperature.

The maximum temperature has a large effect on the etch quality, in terms of selectivity and profile. The photoresist mask suffers from excessive degradation at high temperatures; above the glass transition temperature (T_g) of the photoresist, step increases in the etching rate have been observed [Pons *et al.*, 1991]; in this case, for HPR 204 (a novolak photoresist), T_g was in the range of 80-100°C.

The effect of He pressure on the ultimate wafer temperature ($T_{w\infty}$)is shown in Figure 6-9, for an 87 SCCM Cl_2 discharge operating at 300 mTorr, 475 Watts (387 Watts actual), 0 Gauss, and with a cathode pedestal temperature (T_P) of 20°C. At He pressures above 6 Torr, $T_{w\infty}$ does not change significantly; $T_{w\infty}$ is approximately 28°C above T_p. As the He pressure is dropped, $T_{w\infty}$ increases. In the absence of He cooling ($p_{He} = p_{chamber}$), $T_{w\infty}$ 138°C, about 118°C above T_p. This is a fairly significant increase. Also shown on this plot is $T_{w\infty}$ - T_p which is the driving force for heat transfer.

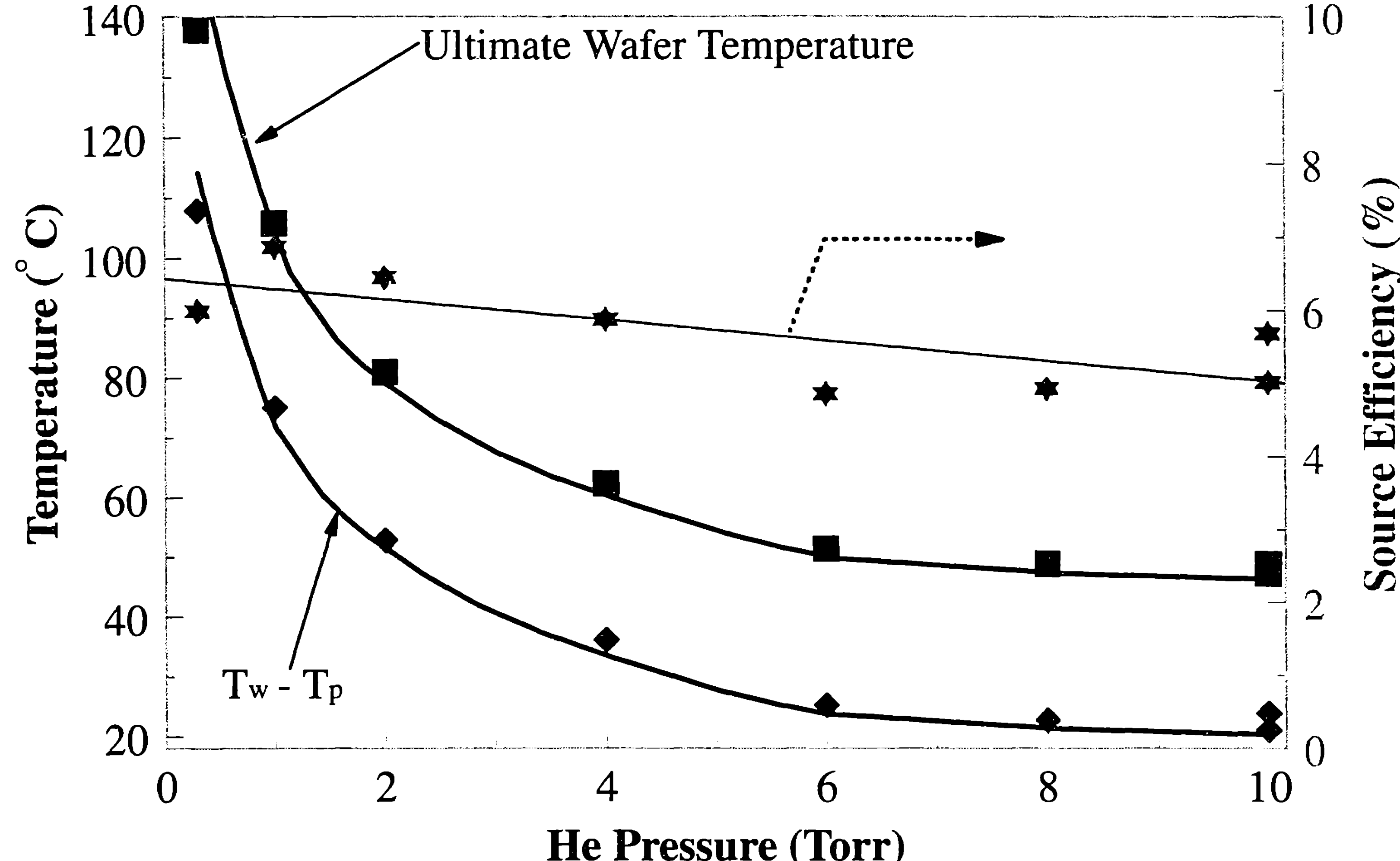


Figure 6-9: Ultimate Wafer Temperature and (T_w - T_p) as a Function of He Pressure. Cl_2, 87 SCCM, 300 mTorr, 475 Watts, 0 Gauss, 20°C Cathode. Also Shown is the Source Efficiency.

The second important parameter is the rate of temperature rise; plotted in Figure 6-10 are T_w, T_p, and T_w - T_p as a function of time for a 500 Watt, 0 Gauss, 100 mTorr, 30 SCCM Cl_2, 10 SCCM HBr plasma. The cathode chiller was set to 20°C, and the He pressure was 4 Torr. Initially, T_w shows a rapid increase (3.3 °C/sec) due to heating of the wafer which is a very small thermal mass. After this rapid rise, T_w increased slowly, in this case at 0.046°C/sec. This long slow increase is due to an increase in Tp, which is due to the construction of the cathode. The cathode assembly shown in Figure 2-14 consists of two separate regions. The lower region has cooling fluid circulating through a series of bore holes. This region is very well cooled and showed almost no measurable temperature variation during etching. The upper region, the cathode pedestal, is simply bolted onto the cathode base. These is no provision to increase the heat transfer between the two regions. For this reason, the temperature of the pedestal increases slowly during wafer processing. To maintain T_p closer to the chiller temperature, T_c, the heat transfer between the pedestal and the base needs to be improved, possibly with something such as diffusion pump oil.

The two components that make up the wafer temperature rise are shown more clearly in Figure 6-11, which shows T_w (from Figure 6-10) and the two components. The rapid exponential increase in the wafer temperature is shown as a dotted line. The component rapidly reaches its ultimate value. The time required to reach this ultimate value can be estimated from the characteristic response time of the wafer, t_c, which is given by

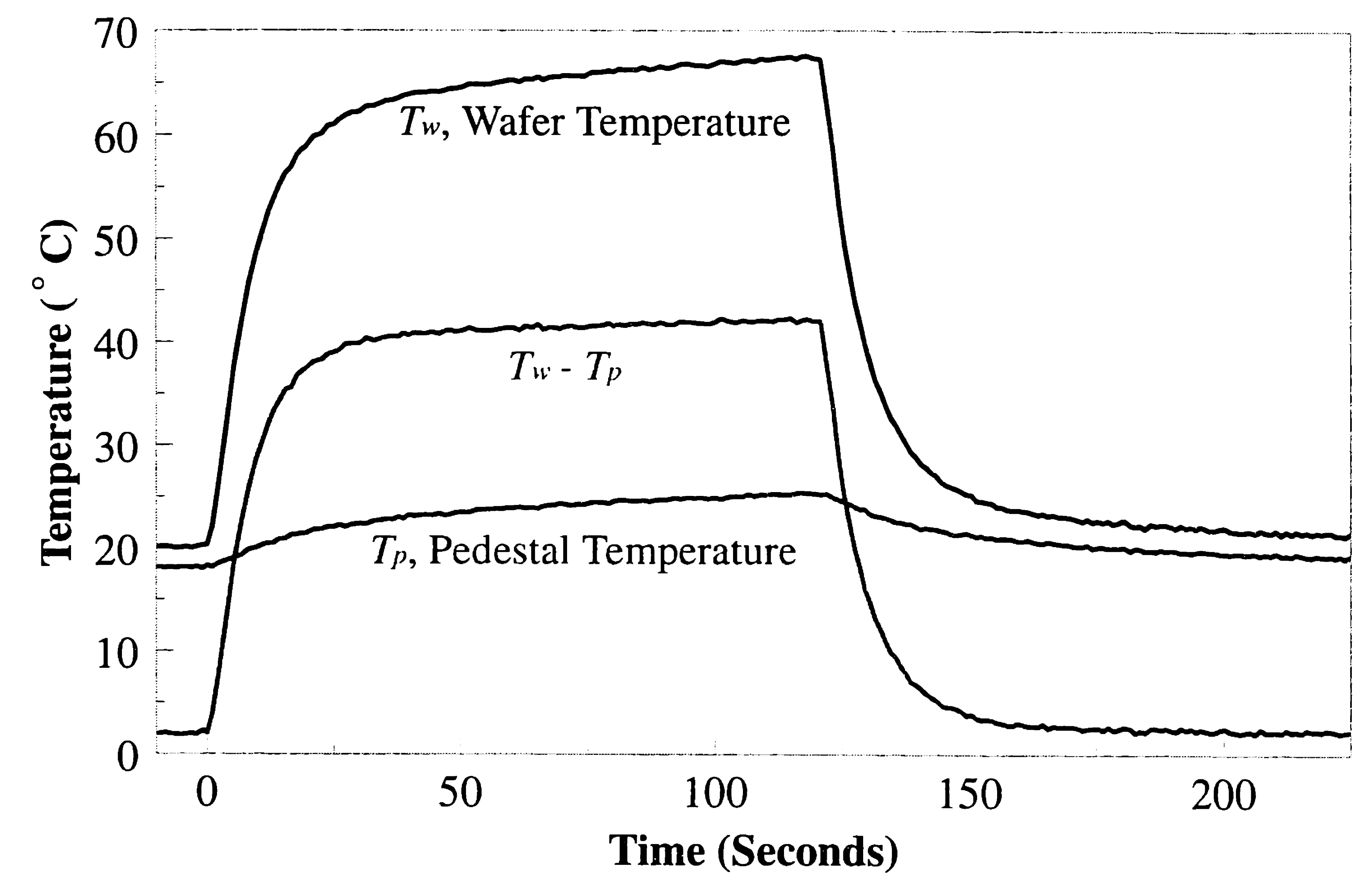


Figure 6-10: Wafer and Pedestal Temperatures and Their Difference as a Function of Time. 500 Watt, 0 Gauss, 100 mTorr, 30 SCCM Cl_2, 10 SCCM HBr, 4 Torr He, 20°C.

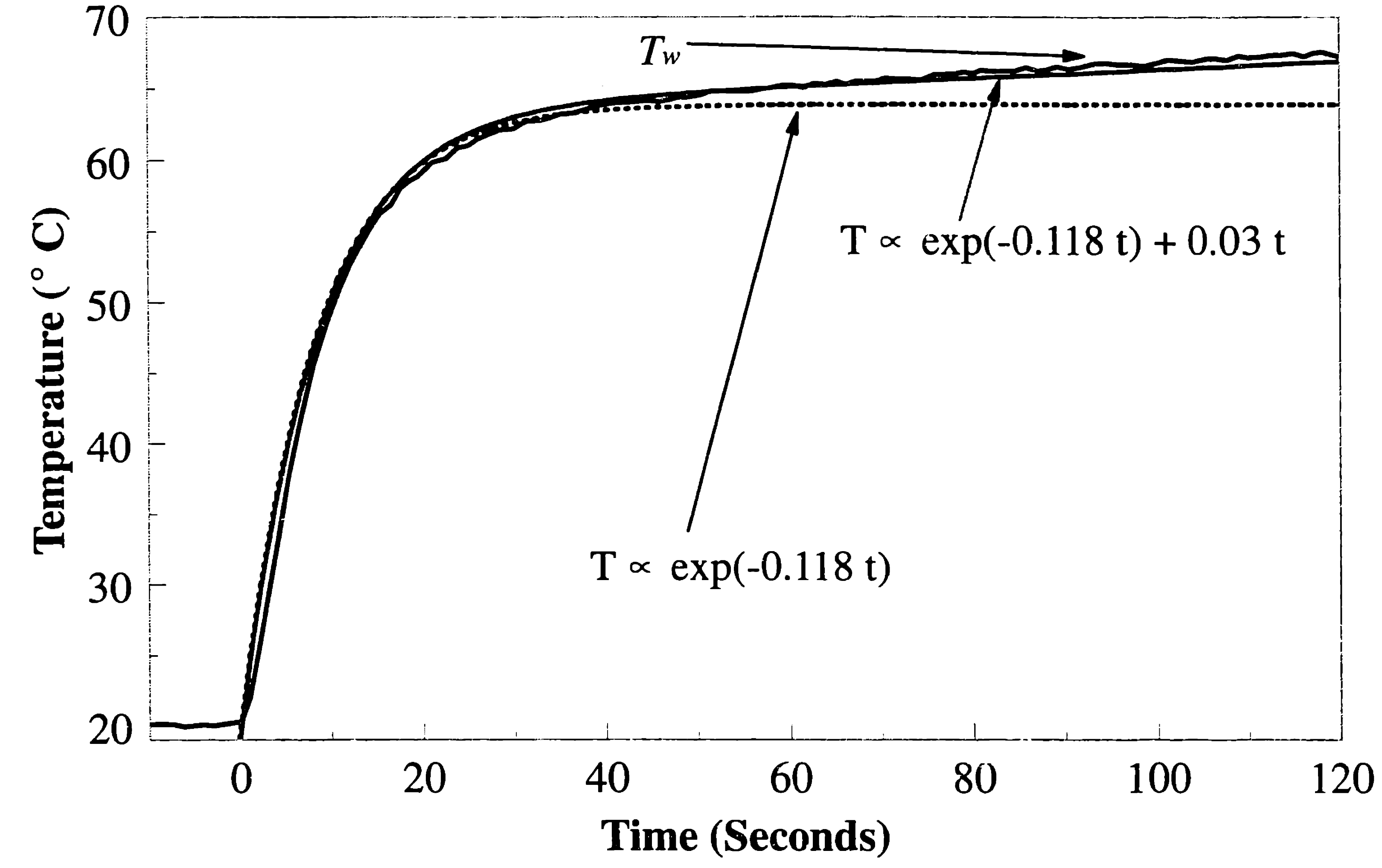


Figure 6-11: Wafer Temperature and Model Fits as a Function of Time. Wafer Temperature Response Consists of a Rapid Exponential Response Due to Wafer Heating and a Slow Linear Response Due to Pedestal Heating.

$$t_c = \frac{\rho c_p a^2}{k_{Si}} = \frac{a^2}{\alpha_{Si}} \quad \textbf{(6-24)}$$

where α_{Si} is the thermal diffusivity of Si (0.92 cm^2 s^{-1}). For a 100-mm diameter wafer, this corresponds to 27 sec. From Figure 6-11, the time required to reach this is seen to be on the order of 30 sec (approximately 3.5 time constants for the exponential rise). This time corresponds to a Fourier Number, $N_{Fo} = t / t_c$, on the order of 1.

T_w continues to increase after this time. The rate of change of T_p was measured and found to be approximately 0.03°C/sec. This slope was used to increase the wafer temperature after the rapid initial rise. The resulting fit comprised of an exponential increase followed by a linear rise does a good job of modeling the experimental data as shown in Figure 6-11.

The wafer response times that have been published by others vary greatly. Visser [1989] observed the exponential rise in wafer temperature lasting 3-4 minutes for an unclamped wafer with no He cooling. Patel *et al.* [1991] observed transient times on the order of 100-200 seconds. Donnelly *et al.* [1992] saw exponential response times on the order of 1-2 minutes, with a long linear rise in temperature. In this case, the linear rise was much larger in magnitude than the exponential rise. They also attributed the linear rise to the increase in temperature of their clamping assembly/block which was bolted to the chilled electrode base.

6.3.8 Wafer Power Deposition and Ion Current

Wafer temperature measurements combined with other simple measurements can yield much more information than just temperatures. Contained within these measurements are the efficiency of the plasma (in terms of input power deposited at the wafer) and the ion current to the wafer. Derivations for these two parameters are given below.

The heat flux from the wafer is given by Equation (6-3). There are a number of possible sources of heat input to the wafer: heat of reaction from etching reactions, heat of recombination of etchants on the surface, and energetic ion flux (Equation 6-1). Similarly there are a number of possible heat sinks from the wafer: convection to the etchant feed gas, conduction across the wafer backside (Equation 6-3) and radiation (Equation 6-2). A detailed model of heat effects for etching in a Cl_2 plasma has been developed by others [Aydil and Economou, 1993].

In a non-reactive gas such as Argon, the heat of reaction and heat of recombination terms go to zero. Thus, the only heat input is from energetic ion bombardment of the wafer. The efficiency of the etcher is given by the ratio of the power arriving at the wafer, Q_w, to that being dissipated in the plasma, P_{RPM}, as measured with the Comdel RPM. Thus, the efficiency of the plasma source, η_p, can be found from a combination of wafer temperature measurements and electrical measurements; the percent efficiency is given by

$$\eta_p = \frac{Q_w}{P_{RPM}} * 100\% = \frac{h A (T_w - T_p)}{P_{RPM}} * 100\% \tag{6-25}$$

Radiation and conduction across the wafer backside are the only possible mechanisms for heat loss as shown above. In all cases considered, radiation was a minor component and

was neglected. Thus, a steady state heat balance on the wafer yields

$$J_i \langle E_i \rangle A = h A (T_w - T_p) \qquad \textbf{(6-26)}$$

which is solved for J_i to yield

$$J_i = \frac{h (T_w - T_p)}{\langle E_i \rangle} \qquad \textbf{(6-27)}$$

The only remaining parameter required to estimate the ion current density (flux) is the average ion energy. Because the AME-5000 is a very asymmetric system, the ion energy can be approximated as the magnitude of the RF voltage [Liu *et al.*, 1990], which is measured with the RPM. This neglects the contribution of the plasma potential which is small (~20 Volts) relative to the RF voltage (~300 Volts). Thus, using only wafer and pedestal temperature measurements and the RF voltage and plasma power from the RPM, the current density and source efficiency can be estimated using the correlation for h given in Equation 6-11.

Estimates for J_i as a function of magnetic field strength at pressures of 10, 100, and 250 mTorr are shown in Figure 6-12. The two obvious trends in this data are that the ion current increases with magnetic field strength and with pressure. As the magnetic field is increased, the ion density increases, which in turn increases the ion current. Increasing the pressure increases the gas density, which also increases the ion density. Thus, the ion current increases. These predicted ion currents are on the order of 1.0 mA cm^{-2}. A rough estimate to see if these numbers are the correct order of magnitude or not is given by the Child-Langmuir law [Chapman,1980] which predicts J_i as

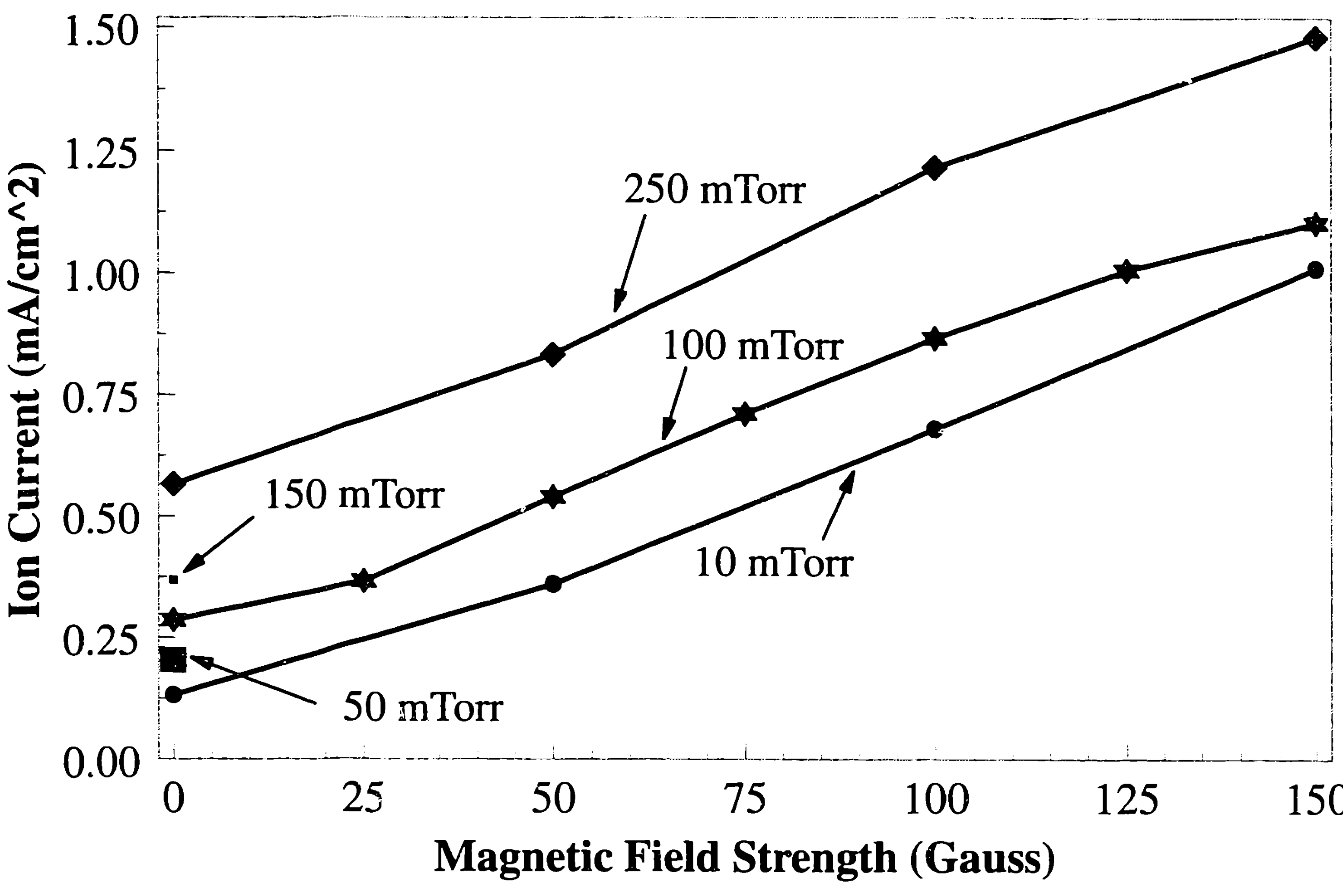


Figure 6-12: Ion Currents Estimated From Tw-Tp for a 250 Watt Ar Plasma.

$$J_i = \frac{4\epsilon_0}{9}\left(\frac{2e}{m_i}\right)^{0.5}\frac{V^{1.5}}{l_{sh}^2} \qquad \textbf{(6-28)}$$

where ϵ_0 is the permeativity of free space $(36\ \pi \times 10^9\ \text{f/m})^{-1}$, e is the electronic charge, m_i is the ion mass, V is the RF voltage, and l_{sh} is the sheath thickness. Using $V = 300$ Volts RMS, and $l_{sh} = 2.5$-mm (Chapter 7), J_i is estimated to be 1.2 mA cm^{-2}. This agrees surprisingly well with the values estimated from temperature measurements.

Estimates of η_p as a function of magnetic field strength for pressures of 10, 100 and 250 mTorr are shown in Figure 6-13. At 0 Gauss, where the AME-5000 is operating in RIE mode, the efficiency ranges from 10% at 250 mTorr to 2.3% at 10 mTorr, indicating that as pressure is reduced, it is harder to couple power efficiently into the plasma. The presence of the magnetic field greatly enhances the source efficiency. At the extreme field strength of 150 Gauss, the efficiency ranges from 20.3% at 250 mTorr to 14.7% at 10 mTorr.

Similarly to Figures 6-12 and 6-13, estimates of J_i and η_p as a function of the nominal plasma power (measured at the RF generator) are shown in Figures 6-14 and 6-15. The power has a very small effect on the ion current (Figure 6-14), when compared to the effect of magnetic field (Figure 6-12). J_i ranges from 0.19 mA cm^{-2} at 100 Watts to 0.57 mA cm^{-2} at 600 Watts. For comparison, the effect of magnetic field is shown at 250 Watts where J_i increases from 0.29 mA cm^{-2} at 0 Gauss to 1.10 at mA cm^{-2} at 150 Gauss. Over a wide range of powers from 100 to 400 watts, η_p is invariant to power, at around 6%. Only at higher powers does η_p vary; it increases to around 13.8% at 600 Watts.

A pseudo-source efficiency has been calculated by Wright *et al.* [1992] using the

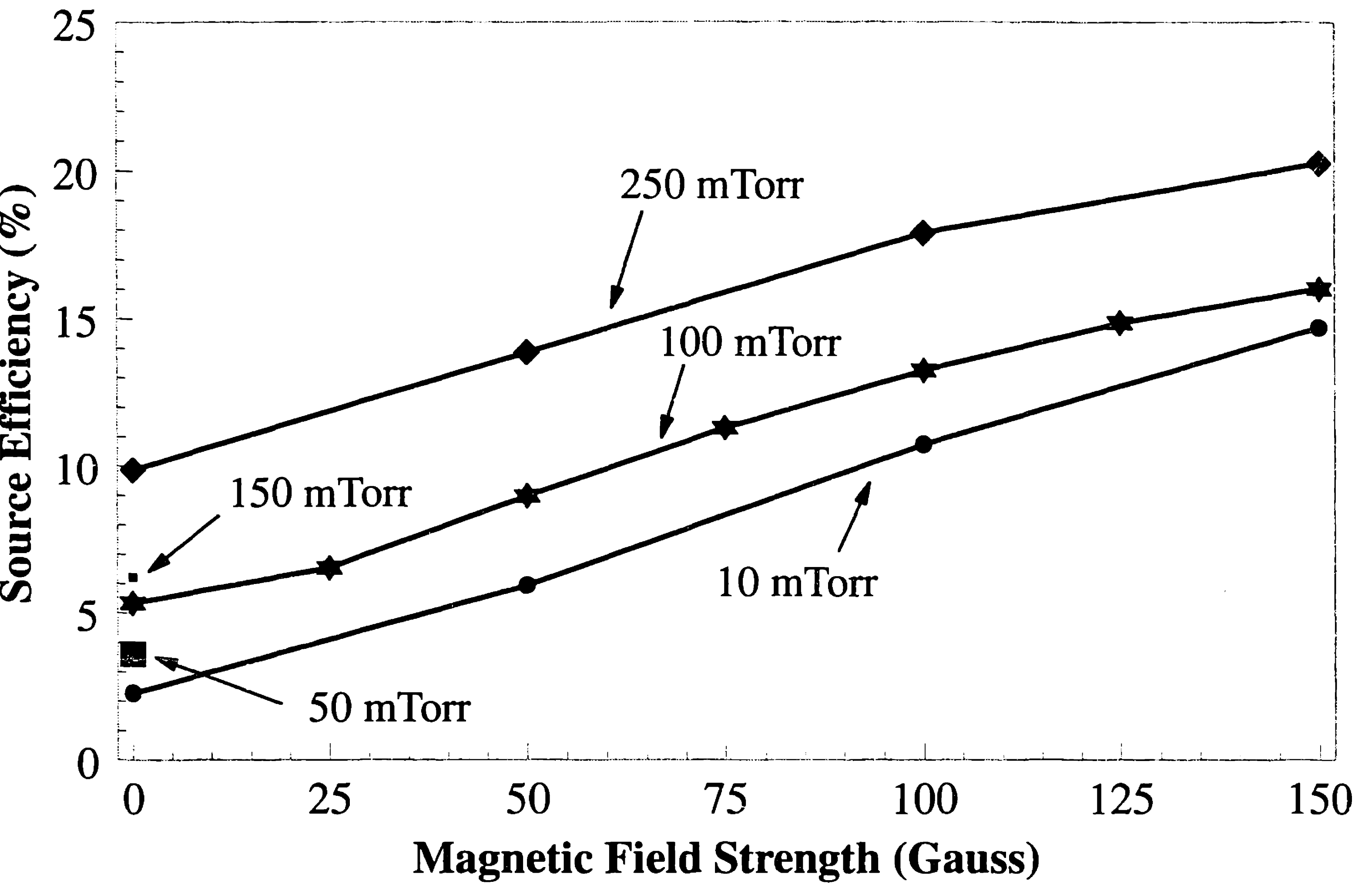


Figure 6-13: Source Efficiency (Power Deposited in Wafer as a Percent of RPM Measured Plasma Power) Estimated From Tw-Tp for a 250 Watt Ar Plasma.

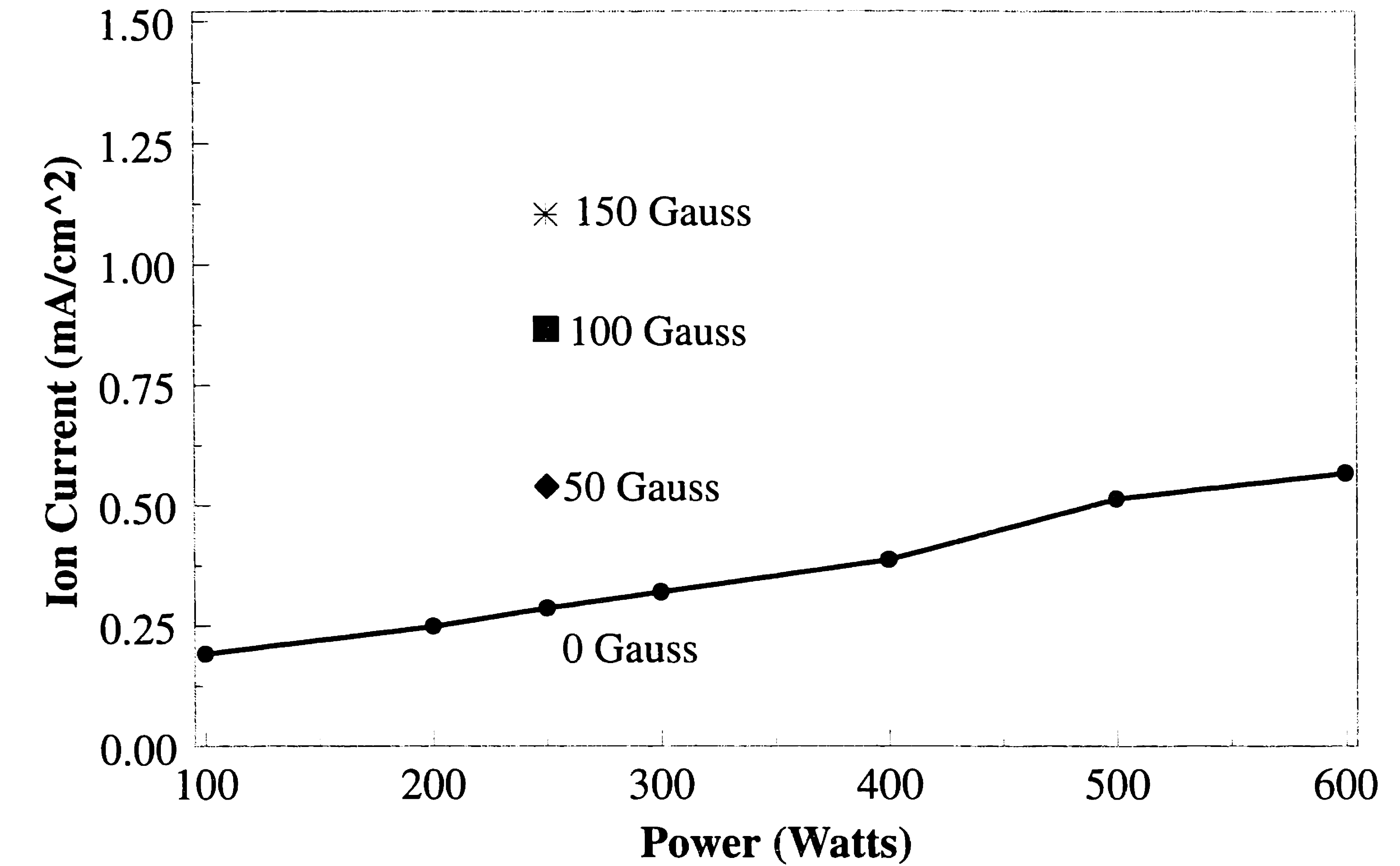


Figure 6-14: Ion Currents Estimated From Tw-Tp for a 100mTorr Ar Plasma. Curve is for 0 Gauss. Points shown for 50, 100, and 150 Gauss at 250 Watts

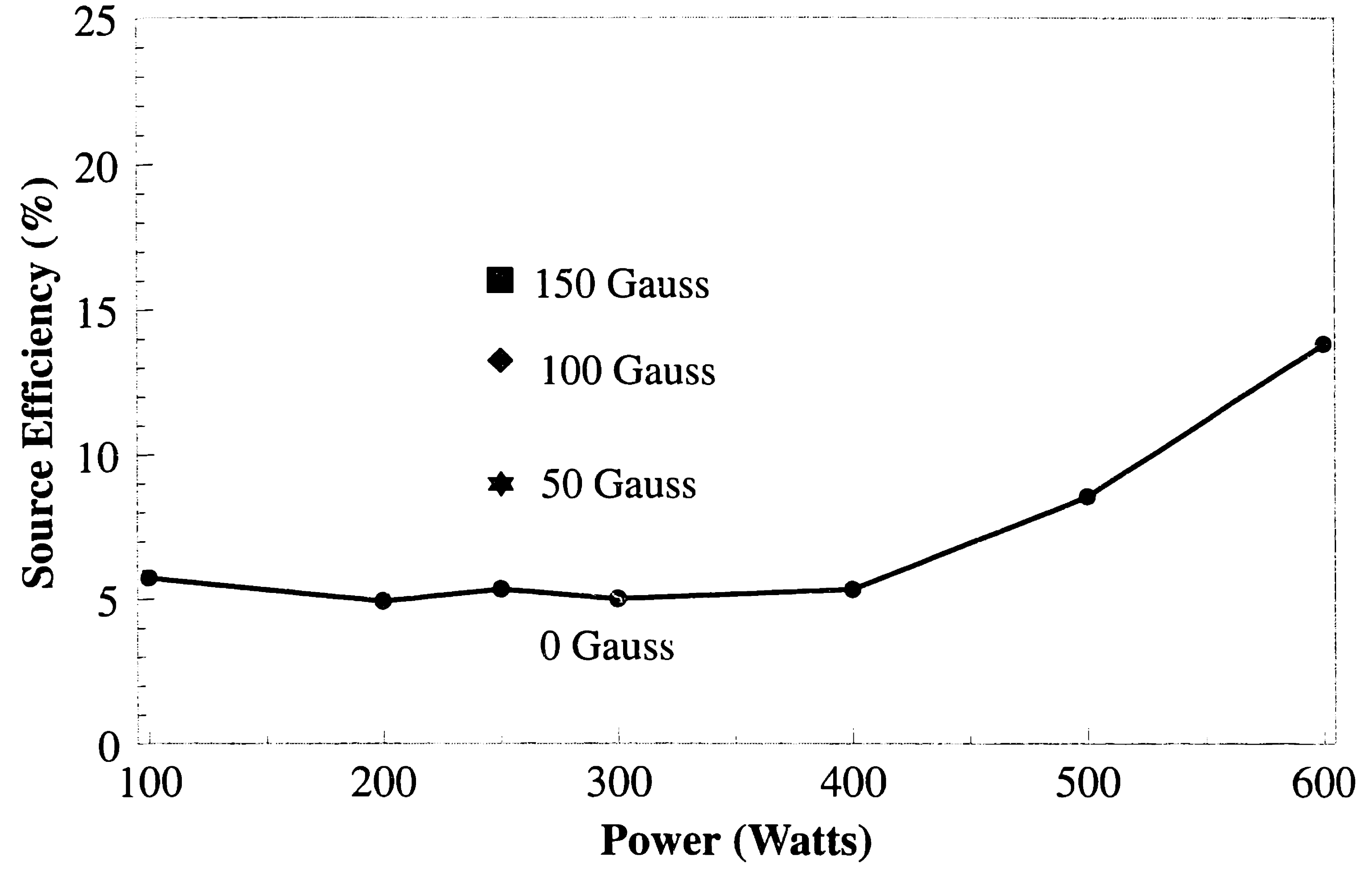


Figure 6-15: Source Efficiency (Power Deposited in Wafer as a Percent of RPM Measured Plasma Power) Estimated From Tw-Tp for a 100mTorr Ar Plasma. Curve is for 0 Gauss. Points shown for 50, 100, and 150 Gauss at 250 Watts

power measured at the RF generator and not the plasma power measured at the cathode. In the case of RIE, they obtained values of η_p on the order of 25%. These values will increase to perhaps 35% when the actual plasma power is considered. Their data also show an increase in η_p with plasma power. However, η_p depended on the He pressure, with lower values of η_p at lower He pressures. When run as a remote source, the triode etcher yielded $\eta_p \sim 5.5\%$.

The effect of He pressure on the estimates of J_i and η_p is shown in Figure 6-16. Over a wide range of He pressure, the estimates are constant. Only at the lowest He pressures do the calculated values of J_i and η_p start to change (notice the greatly zoomed-in scale on the y-axis of this graph in comparison to Figures 6-12, 6-13, 6-14, and 6-15). All of the measurements in this section were done using 10 Torr of He. In an Ar plasma, the source efficiency ranged from 4.5% at 5 Torr to 4.8% at 10 Torr and 5.1% at 0.3 Torr He, with a He pressure-averaged efficiency of 4.7%. The source efficiency is also a function of the gas used. For a 0 Gauss, 300 mTorr, 250 Watt Ar plasma, the above data predicts η_p of around 11%. Increasing the power to 475 watts would be expected to increase η_p further. However, Figure 6-9 shows the source efficiency for a 475 Watts, 30 mTorr, 0 Gauss Cl_2 plasma, which has a He pressure-averaged efficiency of 5.7%, which is significantly less than predicted from the Ar measurements.

6.4 Reduced Temperature Polysilicon Etching in Cl_2

6.4.1 Etch Rates

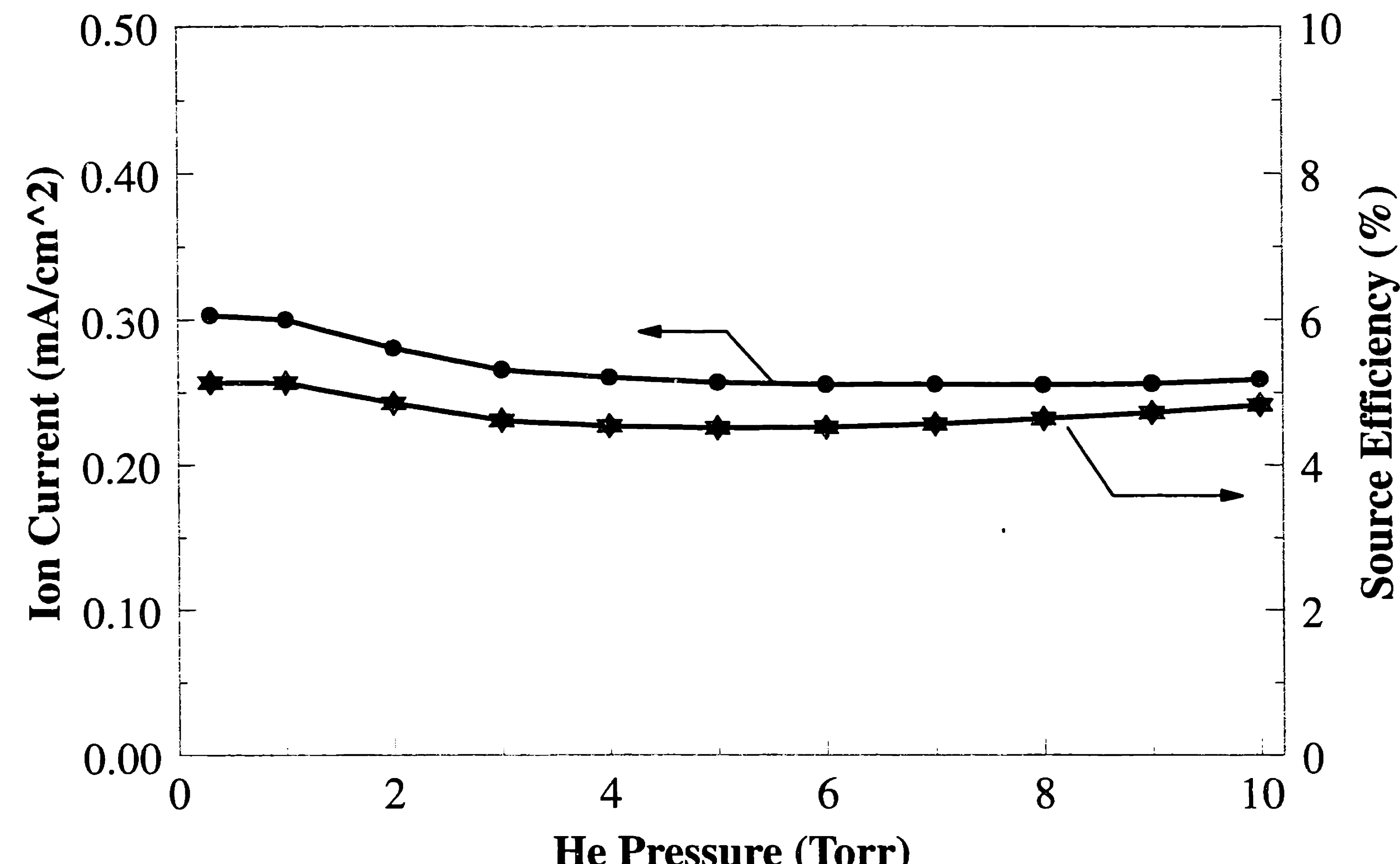


Figure 6-16: Stability of Ion Currents and Source Efficiency Estimated From Tw-Tp for a 250 Watt, 100 mTorr, 0 Gauss Ar Plasma.

A series of several different wafers was etched at a number of different temperatures. The etching conditions were: 40 SCCM Cl_2, 100 mTorr, 250 Watts, 75 Gauss at 2 Hz rotation, and 8 Torr He. Samples of undoped unpatterned poly, undoped patterned polysilicon, and heavily phosphorus doped (13 Ω/□) unpatterned polysilicon were etched at chiller temperatures between -75°C and +20°C. The actual wafer temperature was measured during etching and ranged from -60°C to +35°C. This actual wafer temperature was used instead of the nominal chiller temperature for all calculations.

The etching rate of these three different series of wafers is plotted as a function of the wafer surface temperature in Figure 6-17. Several observations can be made from this data. First, unpatterned undoped poly etches faster than the undoped patterned poly. The photoresist reduces the poly etch rate, either by reactant consumption, reactant recombination, or redeposition (Chapter 5). Second, undoped polysilicon etches faster than doped polysilicon. This is contrary to what is expected. In molecular Cl_2, the etch rate is a strong function of the phosphorus doping level [Ogryzlo *et al.*, 1988]. Finally, all three show a large decrease in the etch rate around the -40 to -60°C range. This decrease was also seen by Richards [1986] in a parallel plate plasma etcher; in that case, the drop occurred at a much higher temperature of 25°C.

The etching rate of silicon by halogens is best fit to a modified Arhenius expression where the preexponential factor is a function of $T^{0.5}$; this has been observed with Fluorine [Flamm and Donnelly, 1981], and Chlorine [Ogryzlo *et al.*, 1988],[Ogryzlo *et al.*, 1990]. For this reason, we fit the etch rate data to a modified Arhenius expression of the form

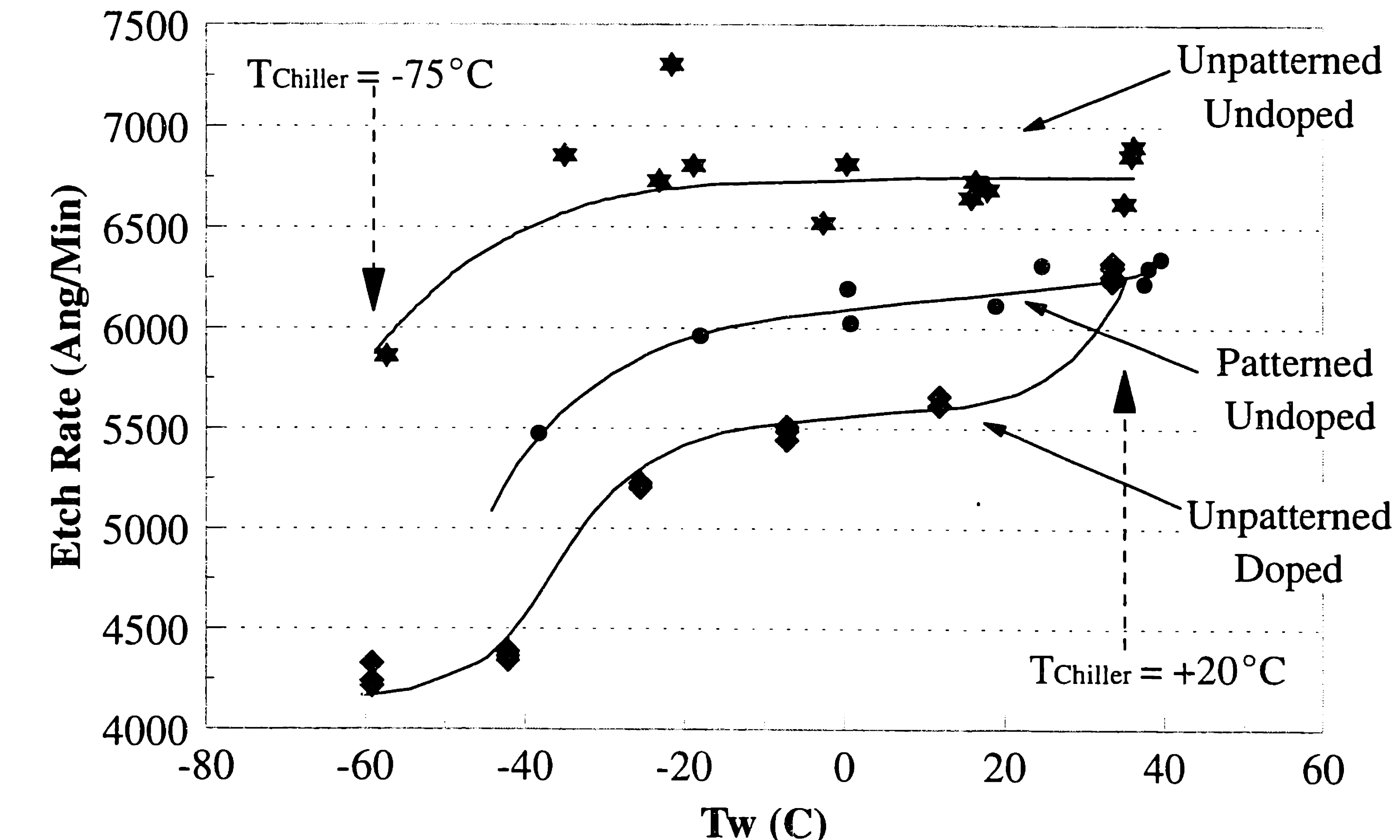


Figure 6-17: Etch Rate for Blanket and Patterned Undoped Poly and n++ (P)-Doped Blanket Poly as a Function of the Measured Wafer Surface Temperature During Etching

$$ER = A\sqrt{T}\exp\left(\frac{-E_a}{RT}\right) \qquad \textbf{(6-29)}$$

where ER is the etch rate, A is the preexponential factor, T is the absolute temperature, E_a is the activation energy, and R is the gas constant. Indeed, the correlation coefficient for the data was better when $A \propto T^{0.5}$ then when $A \neq A(T)$. The data are replotted in Figure 6-18 in the linearized form of $ER*T^{0.5}$ as a function of $1/T$, along with the best fit lines to the data. The activation energies were determined from these data and are given in Table 6-2.

Table 6-2

Activation Energies for Cl_2 MERIE of Polysilicon

Wafer Type	E_a (Kcal/Mol)	E_a (eV/Molec)	Temp. Range (°C)
Doped, Unpatterned	0.81	0.035	-60 - +35
Undoped, Patterned	0.50	0.022	-40 - +35
Undoped, Unpatterned	0.36	0.015	-60 - +35

These activation energies appear to be lower than those determined with chemical etchants (in the absence of ion bombardment) and in other etchers. Data from a number of published papers is plotted in the form of ER vs $1/T$ in Figure 6-19. The data are the above data, along with data on: Cl_2 etching of heavily Phosphorus-doped poly in RIE and magnetron etchers [Sekine *et al.*, 1988]; $Cl_2/SiCl_4/O_2$ etching of heavily phosphorus-

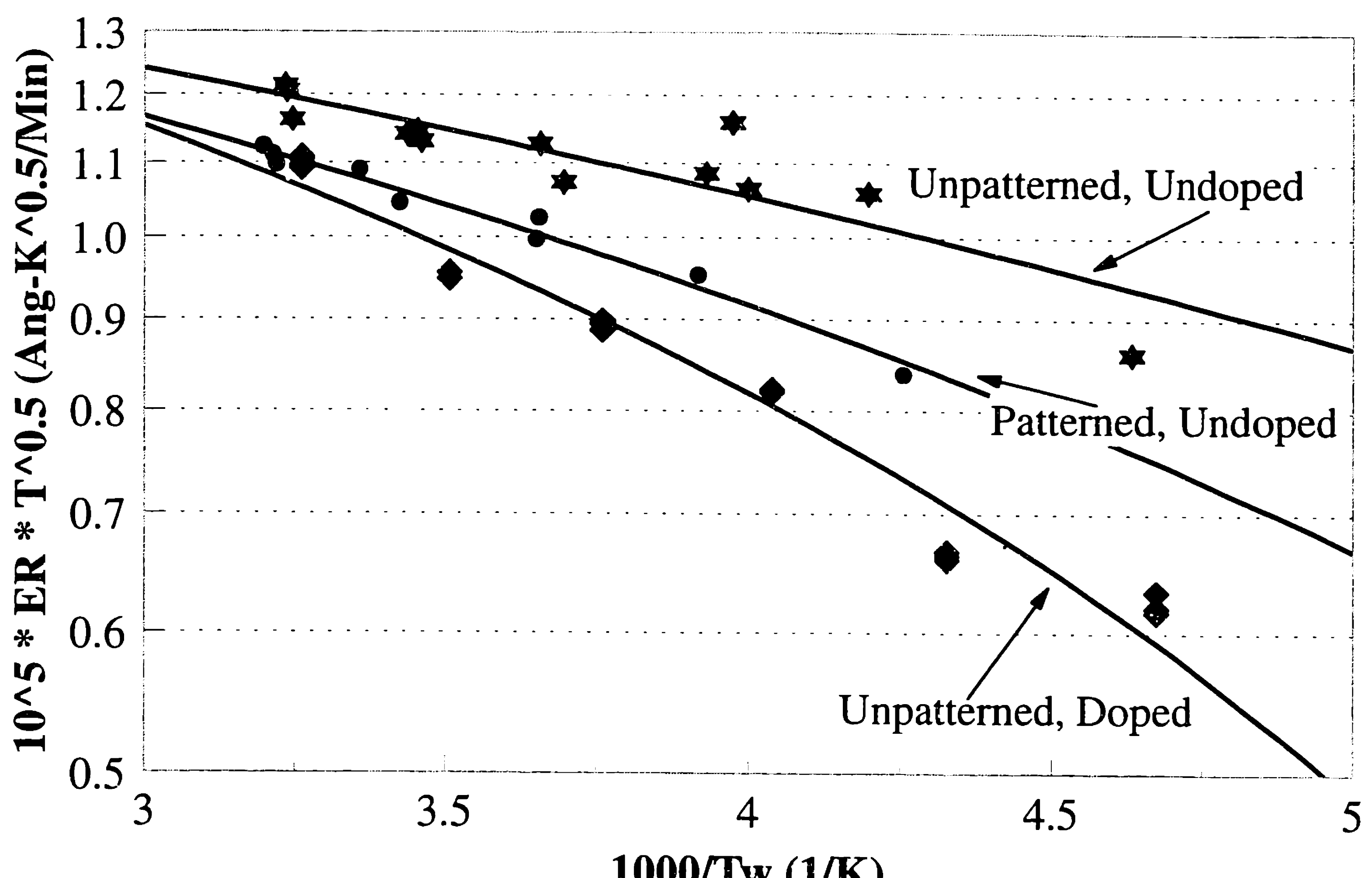


Figure 6-18: Etch Rate*T^0.5 for Unpatterned and Patterned Undoped Poly and n++ (P)-Doped Unpatterned Poly as a Function of the the Inverse of the Measured Wafer Surface Temperature During Etching

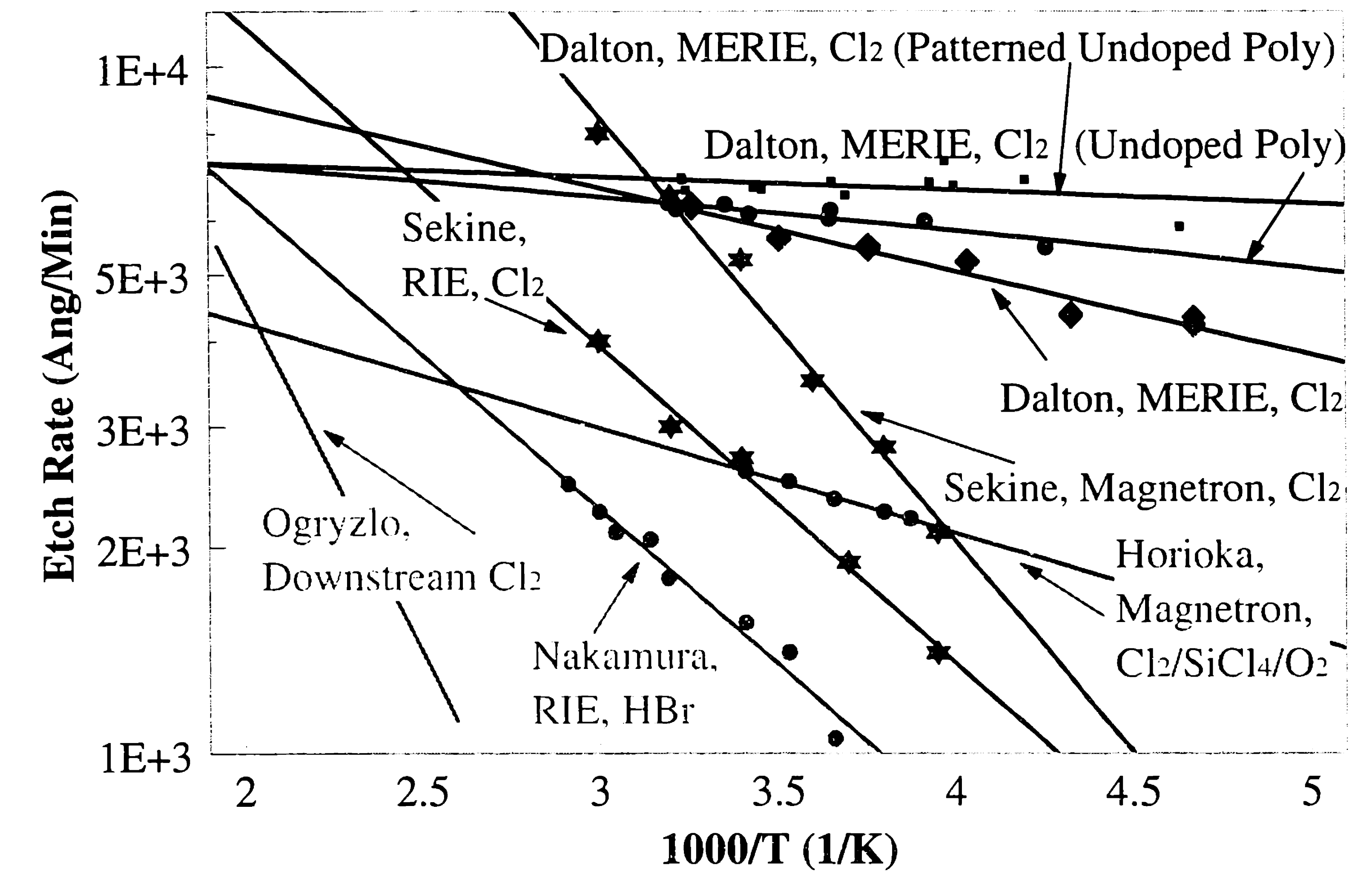


Figure 6-19: Comparison of Temperature Dependance of Etching Rates for Phosphorus Doped Polysilicon Using Various Chemistries and Etchers. References Given in Text.

doped poly in a magnetron [Horioka et al., 1988]; HBr etching of 60 Ω/□ phosphorus-doped poly in a RIE [Nakamura *et al.*, 1988]; and downstream microwave Cl_2 etching of heavily phosphorus-doped poly [Ogryzlo *et al.*, 1990]. The activation energies determined from these data (using a rate model as given in Equation 6-29) were determined and are listed in Table 6-3.

Table 6-3

Activation Energies From Other Researchers

Chemistry	Etcher	E_a (Kcal/Mol)	Reference
Cl_2	Downstream Microwave	4.7	Ogryzlo, 1990
Cl_2	Magnetron	3.1	Sekine, 1988
Cl_2	RIE	2.4	Sekine, 1988
HBr	RIE	2.37	Nakamura, 1988
$Cl_2/SiCl_4/O_2$	Magnetron	0.97	Horioka, 1988
Cl_2	Plasma Etch	1.2	Richards, 1986
Cl_2 Undoped Poly	Plasma Etch	0	Richards, 1986
F	Downstream	2.5	Flamm, 1981

The low activation energies determined for Cl_2 MERIE of polysilicon on the AME-5000 indicates that the etching is highly ion-enhanced. The spontaneous etching component is very small. An estimate of the spontaneous etch rate of phosphorus-doped Si with Cl atoms is given by [Walker and Ogryzlo, 1991a]

$$ER = 7 \times 10^{7}\, nm\, \mathrm{min}^{-1}\, Torr^{-1} \exp\left(\frac{-6.64\ kcal/mol}{RT} \right) P_{Cl} \qquad \textbf{(6-30)}$$

where ER is the etch rate in nm min^{-1}, R is the gas constant (8314 J/mol-K), T is the absolute temperature, and P_{Cl} partial pressure of Cl atoms in Torr. The discharge used for this work was 0.1 Torr of Cl_2, assuming a 10% dissociation. ER at 300K evaluates to about 100 Å/min, or about 1.7% of the observed etch rate of 6000 Å/min.

The difference in activation energy for doped and undoped polysilicon agrees with the trends observed by Richards [1986]. However, in that case, the etch rate of undoped polysilcion was invariant with temperature. In this case, it has a small but finite temperature difference.

6.4.2 Etching Uniformity and Profile

The etching nonuniformity was measured for Cl_2 etching of undoped blanket polysilicon as a function of temperature. The nonuniformity was determined over the wafer leaving a 0.9 cm exclusion zone at the wafer edge. The nonuniformity, NU, was defined as

$$NU = \frac{ER_{max} - ER_{min}}{\langle ER \rangle} * 100\% \qquad \textbf{(6-31)}$$

where ER_{max} was the maximum etch rate, ER_{min} was the minimum etch rate, and $\langle ER \rangle$ was the average etch rate.

The measured etch rate nonuniformity as a function of the measured wafer temperature during etching is shown in Figure 6-20. Initially, as the wafer temperature is

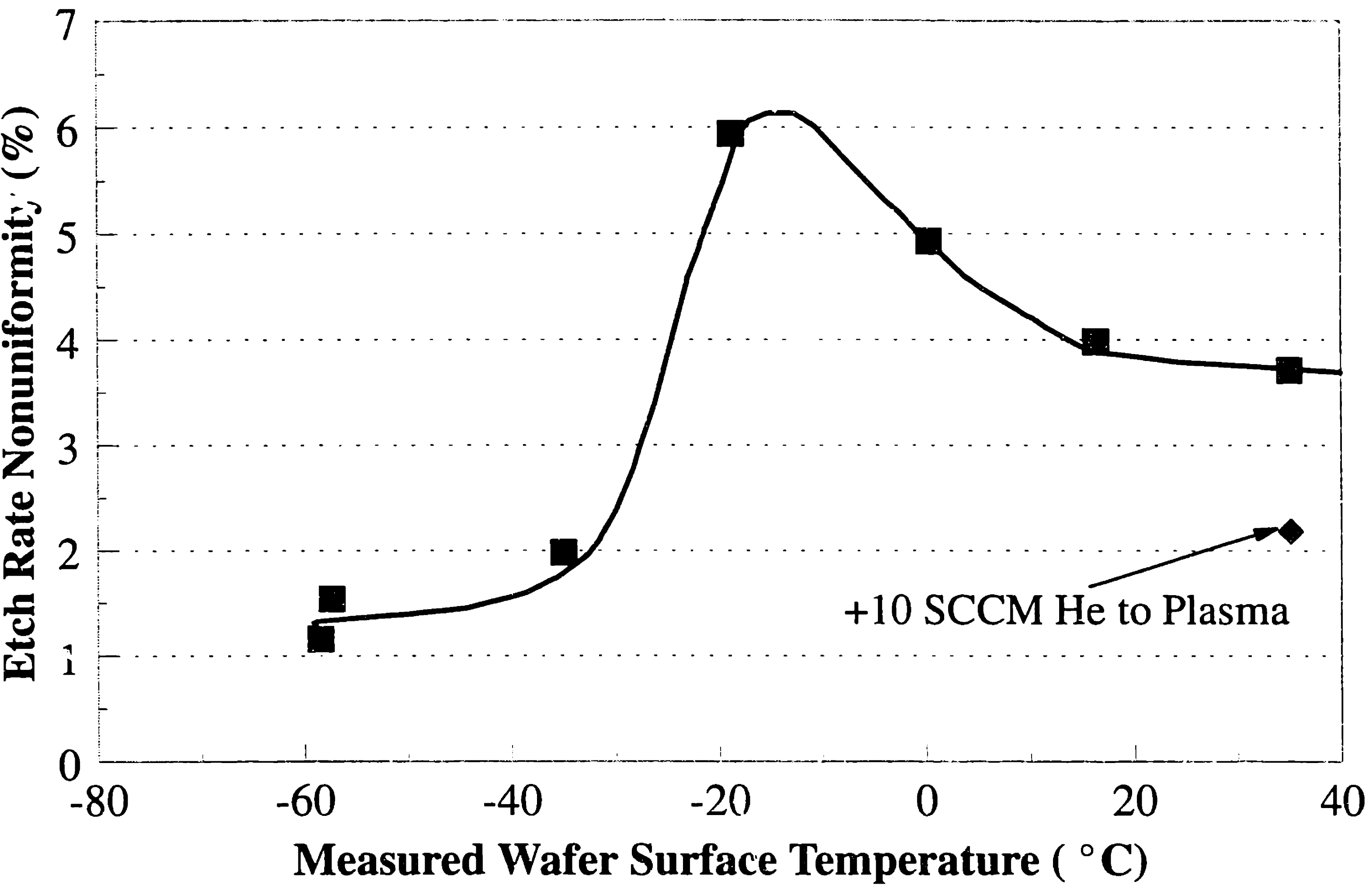


Figure 6-20: Etch Rate Nonuniformity For Undoped Blanket Polysilicon Wafers as a Function of Measured Wafer Temperature During Etching.

reduced from 35°C to -20°C, NU increases from 3.7% to a maximum of 5.9%. As the wafer temperature is decreased further, NU suddenly decreases to 2% at -40°C. NU continues to decrease to around 1.3% as the temperature is reduced further to -60°C. This decrease in NU corresponds to the operating region where the etch rate decreases greatly (Figure 6-17). The effect of temperature on microscopic uniformity (aspect ratio dependent etching) is discussed in detail in Chapter 8.

The dramatic increase in uniformity may be due to the large drop in etch rate. This decrease in the reaction velocity, while the diffusion velocity remains fairly constant (because the gas temperature is fairly constant) leads to a decrease in the Damköhler number (Equation 5-9). The other possible explanation for the increased uniformity is due to a He leak from the wafer backside. At the lowest temperatures (-75°C), the He leaked increased to a few SCCM. Addition of 10 SCCM of He (a very liberal amount compared to the leak) to a +20°C etch yielded fairly uniform etching (Figure 6-20). However, the etch rate increased by about 1000 Å/min over the case with no He. Because an etch rate drop was observed corresponding to the uniformity increase, He is most likely not the cause.

Reduced temperature polysilicon etching was evaluated as a method for obtaining anisotropic etch profiles without the use of sidewall passivation. With the cathode chiller running at +20°C, no undercutting was observed. This could be due to the either: (1) etching of the photoresist as well, so that no undercutting is apparent; or (2) anisotropic etching.

Reducing the temperature changed the slope of the etch profile, with the profile being more sloped at lower temperatures. No undercutting was observed under any

conditions. This is most likely due to the extreme ion-enhanced nature of the etching process as demonstrated by the very low activation energies.

6.5 Conclusions

A Fluoroptic® temperature measuring system was used to monitor wafer and pedestal temperatures during plasma etching. An analysis of heat transfer in the AME-5000 showed that conduction across the wafer backside, through a pressurized He gap was the dominant method of heat transfer from the wafer. The heat transfer coefficient was determined as a function of the He backside cooling pressure; it was on the order of 100 Watts m^{-2} K^{-1}. Deflection of the wafer was found to be on the order of 300 μm at 10 Torr He pressure; this was large enough to decrease the heat transfer coefficient. Doming of the cathode to account for this deflection is a viable method of increasing the heat transfer coefficient and may be needed for larger wafer diameters where the deflection is significantly larger.

Wafer temperature measurements during etching showed that the AME-5000 was characterized by two thermal time constants. There was an initial, rapid response (order of 30 seconds) due to the heating of the wafer that was adequately described by an exponential rise. There was also a slow increase in wafer temperature over a much longer time scale (minutes) due to heating of the cathode pedestal block that was bolted onto the cooled cathode base.

Ion current densities and plasma source efficiency were evaluated as functions of operating conditions; these parameters were derived from temperature measurements.

Ion current densities were on the order of 0.1 to 1.5 mA cm^{-2} and source efficiency (determined as the fraction of the power measured at the cathode that was observed in the wafer) ranged from 2 to 20%. Both Ion current density and source efficiency increased with increasing magnetic field strength, pressure, and nominal power, and were fairly insensitive to the He pressure used to make the measurements.

Etching rates of undoped and heavily phosphorus-doped polysilicon showed very small activation energies (0.36 and 0.81 kcal/mol, respectively) over the temperature range of -60°C to +35°C (as measured during plasma etching). This indicated that the etching process was dominantly ion-enhanced, and not chemical in nature, which was confirmed by the lack of undercutting observed in the etch process. The etch rate nonuniformity initially increased with decreasing temperature, and then rapidly decreased to yield a fairly uniform etch at temperatures around -40 to -60°C. This decrease in nonuniformity corresponded to a large decrease in the etch rate indicating that reactant transport may be leading to nonuniform etching at higher temperatures.

Chapter 7

Spatially-Resolved Plasma and Wafer-Surface Potential Measurements

Wafer surface potentials and plasma floating potentials have been measured in an Applied Materials Precision 5000 plasma etcher using a tuned Langmuir probe. Plasma nonuniformities generated by the presence of the magnetic field through the "E x B" drift of electrons generate variations in the wafer surface potential and in the plasma floating potential in the direction perpendicular to the magnetic field. The magnitude of the variations increases with decreasing pressure, increasing power and increasing magnetic field strength.

7.1 Introduction

Damage of devices and structures on a wafer surface during plasma etching is not a new phenomena. Damage mechanisms include ion bombardment of the wafer and charge buildup [Fonash, 1990]. One particular type of damage is the breakdown of gate oxide layers during the polysilicon gate etch; this has been observed during reactive ion etching (RIE) [Watanabe and Yoshida, 1984], magnetron etching (ME) [Greene *et al.*, 1991],[Arikado *et al.*, 1987], magnetically enhanced reactive ion etching (MERIE) [Shin *et al.*, 1993a], and microwave etching [Kawamoto, 1985].

The degree of gate oxide damage depends on the size of devices being made (the so-called antenna effect), the position of the device on the wafer, the gate oxide thickness,

the process chemistry, and on process conditions. When the size of the polysilicon area being etched on a device is varied (keeping constant the gate oxide area), the degree of gate oxide breakdown increases with larger area ratios (poly to oxide) [Shone *et al.*, 1989]. The effect of device position, oxide thickness and antenna ratio on breakdown, has recently been studied in a barrel-asher, a magnetron, and a electron cyclotron resonance (ECR) etcher [Nojiri and Tsunokuni, 1993]. It is clearly seen that as gate oxides become thinner, the damage increases. Damage from Cl_2 etches has been shown to be lower than mixed Cl_2/SF_6 etching [Gabriel, 1991].

The mechanism of gate oxide damage has been disputed for some time. Early studies indicated that damage did not occur during the initial stages of RIE nor during overetch [Watanabe and Yoshida, 1984]; the theorized mechanism was that charge built up on the wafer during etching and was neutralized by current flow across the gate oxide by charge built up on the blocking capacitor at plasma extinction. This current flow caused the oxide damage. Recent work suggests that gate oxide damage occurs because of plasma nonuniformities [Fang and McVittie, 1993],[Gabriel and McVittie, 1992]. These nonuniformities generate locally non-uniform ion and electron currents (globally they are net neutral). During etching of the poly gate, currents flow through the poly, neutralizing the nonuniformities. However, during the final stages of etching, as the poly film becomes discontinuous, charge build up can occur. Damage is caused by tunneling through the gate oxide. The damage caused by tunneling is expected to become significantly greater as oxide film thickness is reduced due to the exponential nature of the process [Greene *et al.*, 1991].

The degree of gate oxide damage is worse in magnetically enhanced systems (both

ME and MERIE). The reason for the increase in damage is an increase in plasma nonuniformity caused by the magnetic field. A typical etcher configuration is shown in Figure 7-1. The electric field is normal to the wafer surface. A magnetic field parallel to the wafer surface is shown.

The magnetization of the plasma can be understood in terms of two characteristic lengths. The first is the collision mean free path of a particle (j) , λ_j, which is given by

$$\lambda_j = \frac{c_j}{\nu_{jN}} \tag{7-1}$$

where c_j is the speed of particle j, and ν_{jN} is the particle-neutral collision frequency. λ_j is the average distance traveled by particle j before colliding with a neutral gas species. In the presence of a magnetic field, a charged particle gyrates with a characteristic length known as the Larmor radius of particle j, r_{lj}, which is given by

$$r_{lj} = \frac{m_j c_j}{q B} \tag{7-2}$$

where m_j is the particle mass, q is the electronic charge, and B is the magnetic field strength. For a magnetic field to influence particles in the plasma, the condition to be satisfied is $r_{lj} < \lambda_j$; physically this indicates that the magnetic field has caused particle j to move with a path length less than the path length from collisions. This idea is expressed with the magnetization number of particle j, $N_{mag,j}$, which is defined as

$$N_{mag,j} = \frac{r_{lj}}{\lambda_j} \tag{7-3}$$

The condition for magnetization to influence a particles trajectory corresponds to $N_{mag,j}$ <

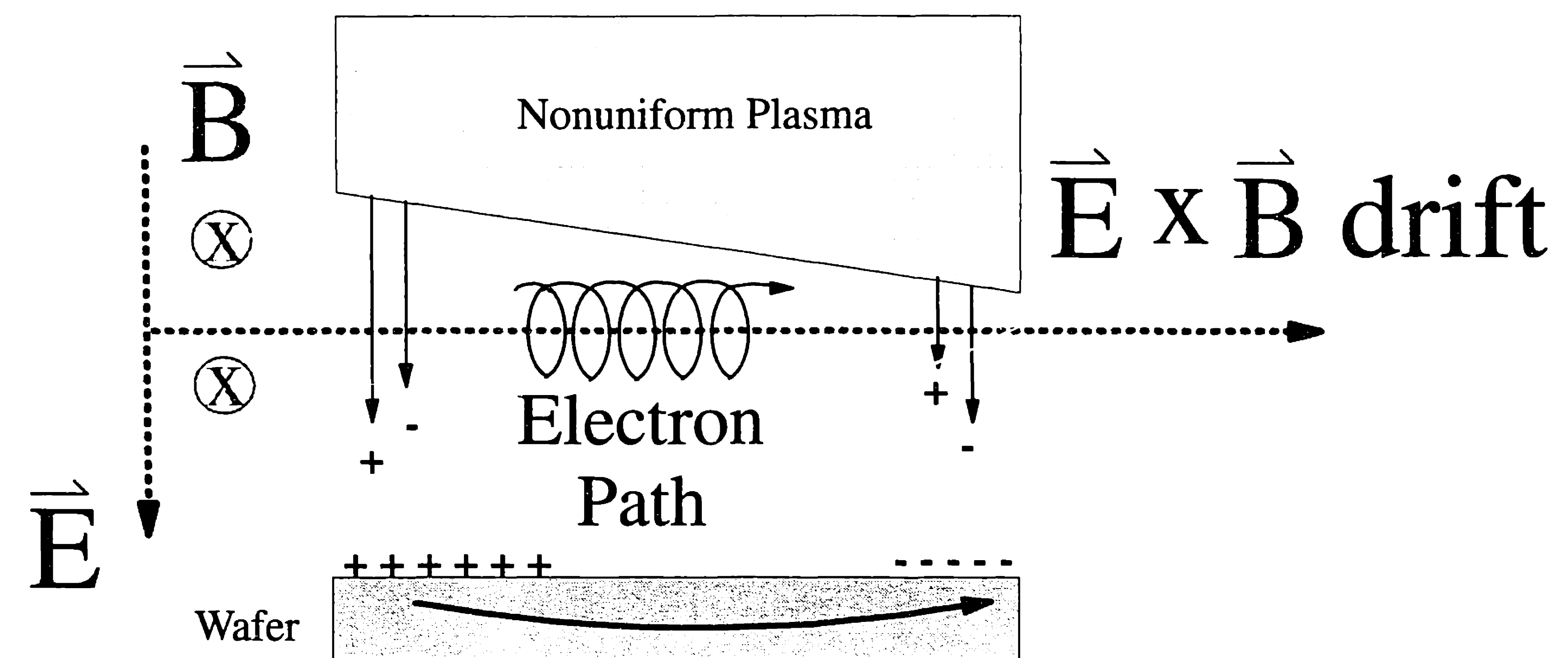


Figure 7-1: Typical MERIE Configuration, Showing Electric and Magnetic Field Directions as Well as the Effective Electron Path, and the Direction of the "E Cross B" Drift of Electrons, Leading the the Production of a Non-Uniform Plasma.

1. Estimates of λ_j, r_{lj}, and $N_{mag,j}$ for a 100 mTorr Cl_2 discharge with an electron temperature of 2 eV and an ion temperature of 0.025 eV as a function of magnetic field strength are listed in Table 7-1.

Table 7-1

Larmor Radii and Mean Free Paths for Electrons and Ions

Mag.Field (Gauss)	$r_{l,Cl}^+$ (cm)	λ_{Cl}^+ (cm)	$r_{l,e}^-$ (cm)	λ_e^- (cm)	$N_{mag,Cl}^+$	$N_{mag,e}^-$
1	96	0.1	3.4	0.62	916	5.5
10	9.6	0.1	0.34	0.62	91.6	0.55
100	0.96	0.1	0.034	0.62	9.16	0.055
1000	0.096	0.1	0.0034	0.62	0.916	0.0055

From Table 7-1, we see that only a small magnetic field strength (5.5 Gauss) is required for $N_{mag,e}^- < 1$, while a much stronger magnetic field (915 Gauss) is required for $N_{mag,Cl}^+ < 1$. The AME-5000 was capable of producing magnetic fields of 0 to 150 Gauss. Thus, under these conditions, the electrons respond to the magnetic field, while the ions do not. Similarly, pressure has a strong effect on N_{mag} as well because pressure changes λ_j by changing the gas density. At 100 Gauss, $N_{mag,e}^- < 1$ for pressures below 1825 mTorr, while $N_{mag,Cl}^+ < 1$ for pressures below 11 mTorr.

The force, $\boldsymbol{F}$, on an electron due to the electric and magnetic fields is given by

$$\boldsymbol{F} = e\boldsymbol{E}(t) + e\boldsymbol{V}\times\boldsymbol{B}(t) \tag{7-4}$$

where e is the electronic charge, $\boldsymbol{E(t)}$ is the time varying electric field, $\boldsymbol{V}$ is electron velocity, $\boldsymbol{B(t)}$ is the time varying magnetic field, and $\times$ indicates the vector cross product. In the configuration of Figure 7-1, electrons gyrate around the magnetic field lines (Larmor motion) [Mitchner and Kruger, 1973]. This increases the effective path of electrons in the plasma, decreases the electron loss rate coefficient and increases the plasma density.

The motion of electrons along magnetic field lines is called "E cross B drift"; it leads to bulk electron motion towards one side of the plasma, which produces a spatially varying plasma density and sheath width, and should produce a spatially varying DC bias on the wafer.

There have been very few attempts to measure the spatially varying DC bias. One attempt used a 1 mm diameter hole in a polyimide film covering the cathode to measure the DC bias at that position [Horiike *et al.*, 1987],[Arikado *et al.*, 1987]. In this case, the DC bias distribution was more uniform in the presence of a magnetic field than without it. The plasma density nonuniformities present without a magnetic field were reduced by the field's presence. When the magnetic field strength was increased from 100 to 400 Gauss, inducing larger variations in the DC bias, an increase in device breakdown was seen. Their results for gate oxide damage correlated well with the measured DC bias variations. Another attempt to measure DC bias variations was made using a series of 9 isolated electrical probes recessed into the cathode [Tsuzuki *et al.*, 1993]. In this case, DC bias variations were seen in the direction perpendicular to the magnetic field, while no variations were seen parallel to the magnetic field.

7.2 Experimental Setup

7.2.1 Etcher

Experiments were conducted on an Applied Materials Precision 5000 plasma etcher (AME-5000), running either as a magnetically enhanced reactive ion etcher (MERIE) or as a reactive ion etcher (RIE) with 100-mm wafers. The wafer was physically clamped onto the cathode by a clamp ring. The anode was the chamber walls and lid. This was a highly asymmetric reactor. A rotatable, variable-strength magnetic field was oriented parallel to the cathode surface (and thus the wafer). Access to the etching chamber was restricted to three side viewports and to one overhead viewport.

7.2.2 Langmuir Probe

A tuned filter was used to measure the DC bias without radio frequency (RF) interference. Instead of using an external tuning circuit as others have done [Paranjpe *et al.*, 1990], a series of carefully chosen RF chokes were used as the tuning circuit [Surendra, 1985]. A number of rf chokes (inductors, J.W. Miller, 9230 Series) were evaluated for use in a Langmuir probe. Figure 7-2 shows the impedance versus frequency for three of the inductors evaluated. The impedance was measured on an HP 9194A impedance analyzer. Similar results were obtained using an HP 3325B function generator and a Lecroy 9400 digital oscilloscope to manually measure impedance. The dashed vertical line on Figure 7-2 indicates the location of 13.56 MHz. Single inductor

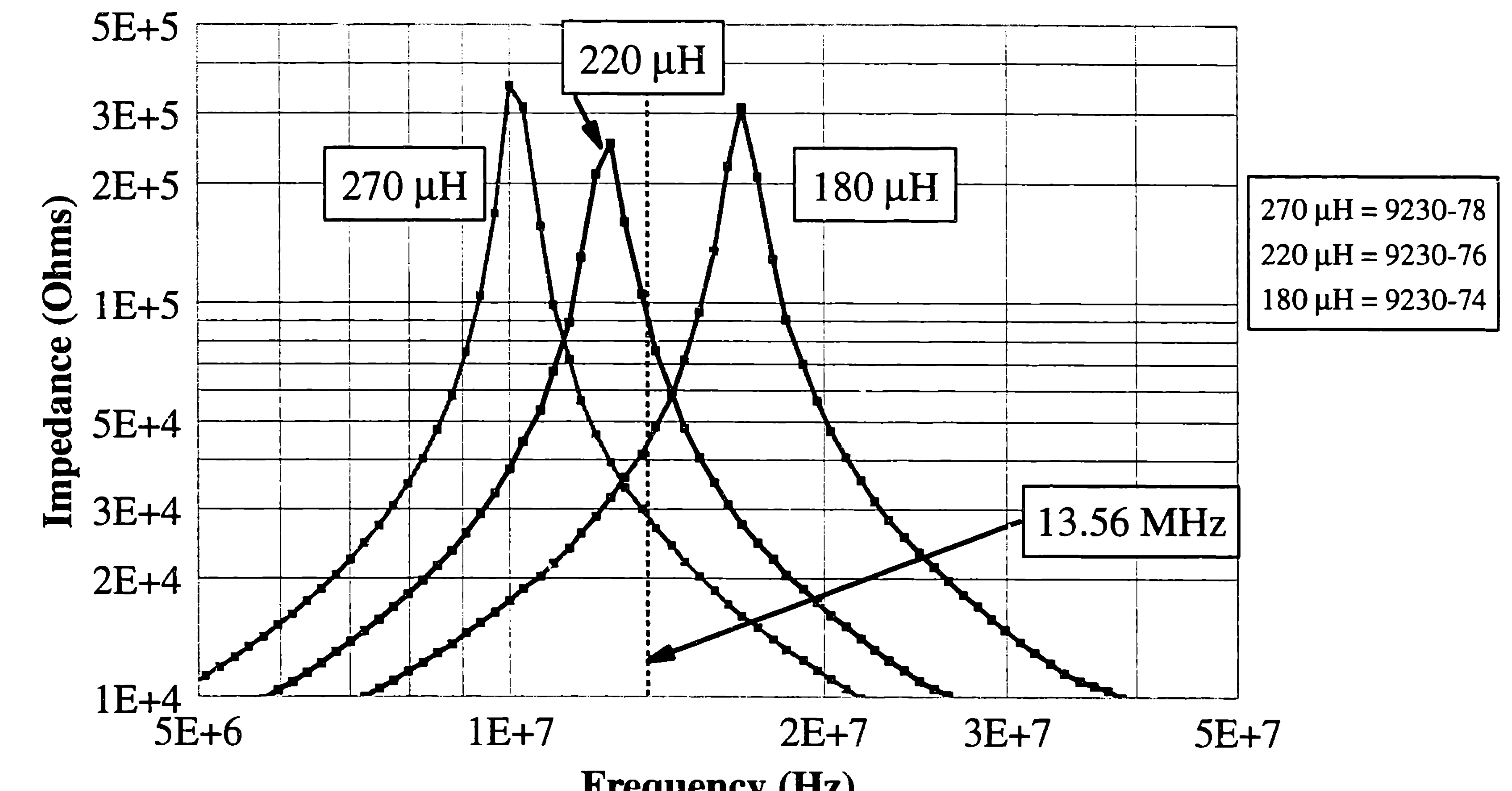


Figure 7-2: Impedance of RF Chokes as a Function of Frequency for a Number of Components. The Self-Resonant Frequency of the Inductors Occurs at the Maximum in Impedance. This Frequency is Shifted by Capacitive Coupling When the Chokes are Arranged in Series.

impedances of 100 kΩ or more could be obtained at 13.56 MHz. The self-resonant frequency for the inductors (as shown in Figure 7-2) was shifted by capacitive coupling between inductors when they were chained together. This shift was beneficial as it moved the self-resonant frequency closer to 13.56 MHz.

The final probe used a series of 9230-76 (220 μH) inductors. The inductors were connected by crimping the leads together inside a 3-mm long section of 0.125 inch O.D. 316 SS tubing. The series of inductors was placed inside a 0.25 inch O.D. section of pyrex tubing which was necked down at one end.

To introduce the probe inside the P5000, a custom lid and flange plate were manufactured. The standard 2-inch optical viewport on the chamber lid was replaced with a 4-inch, o-ring sealed opening. A 316-SS flange plate was placed inside the opening. The Langmuir probe was attached to a linear/rotary motion feedthrough (MDC #672000), and fed vertically into the chamber, off center, as shown in Figure 7-3. From the end of the probe, a section of 10- or 20-mil tungsten wire was run parallel to the wafer surface, isolated from the plasma inside a Teflon® sleeve. This path was chosen to minimize the potential differences the tungsten wire encountered while crossing the wafer, to minimize plasma corruption. The probe tip was bent at the end to facilitate contact to the wafer surface.

The probe feedthrough used necessitated the test wafer design shown in Figure 7-4. In this figure, the probe is fed into the etching chamber vertically above the spot shown on the left side of the wafer. A series of 13 isolated, 0.5 cm x 0.5 cm 10-mil thick metal pads (Al or SS) are mounted onto a silicon wafer covered with 2.0 μm of SiO_2. The area of the metal pads did not change the results. The probe sweeps out the

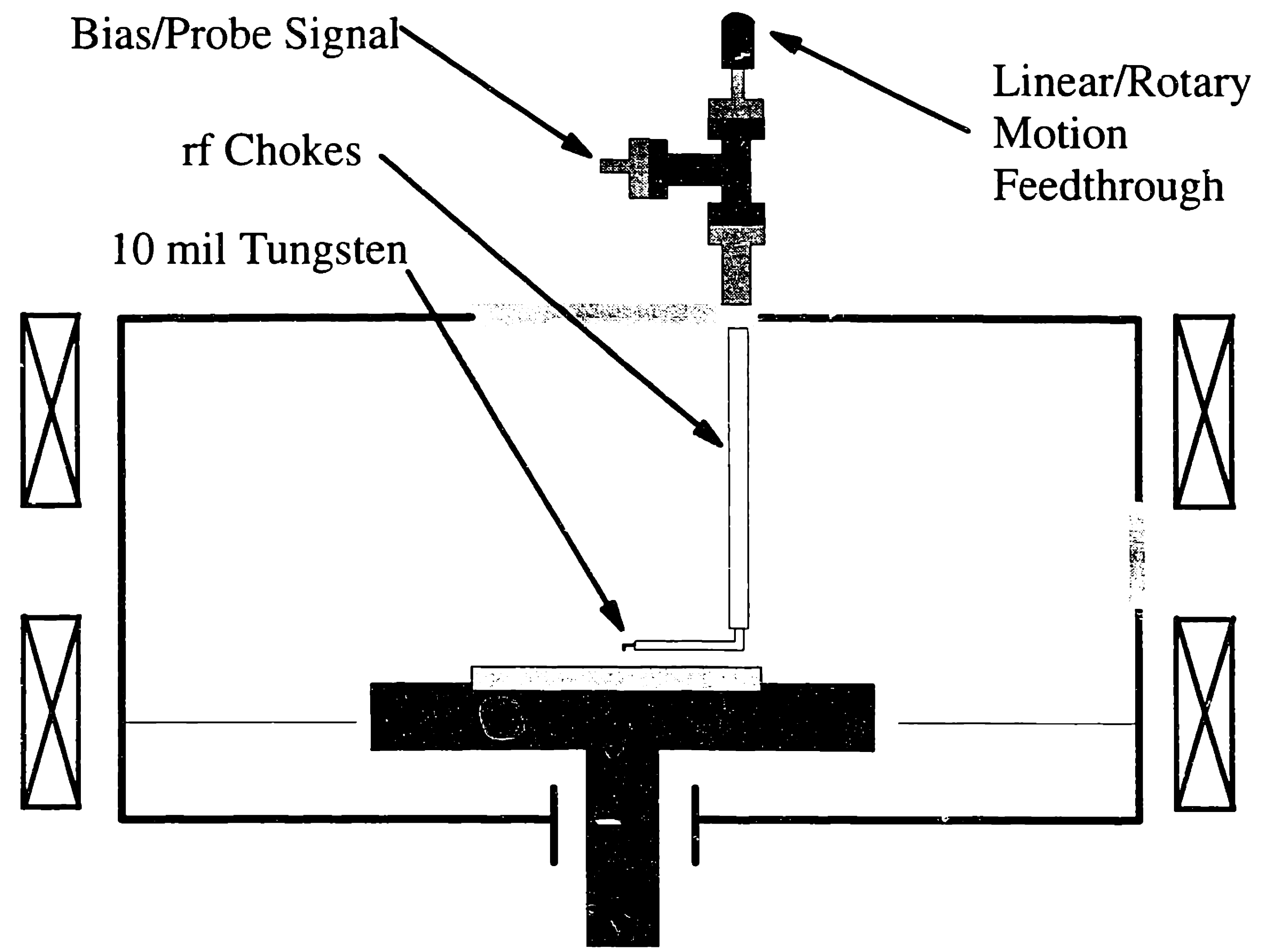


Figure 7-3: Schematic View of Langmuir Probe Feedthrough and Positioning on Etcher.

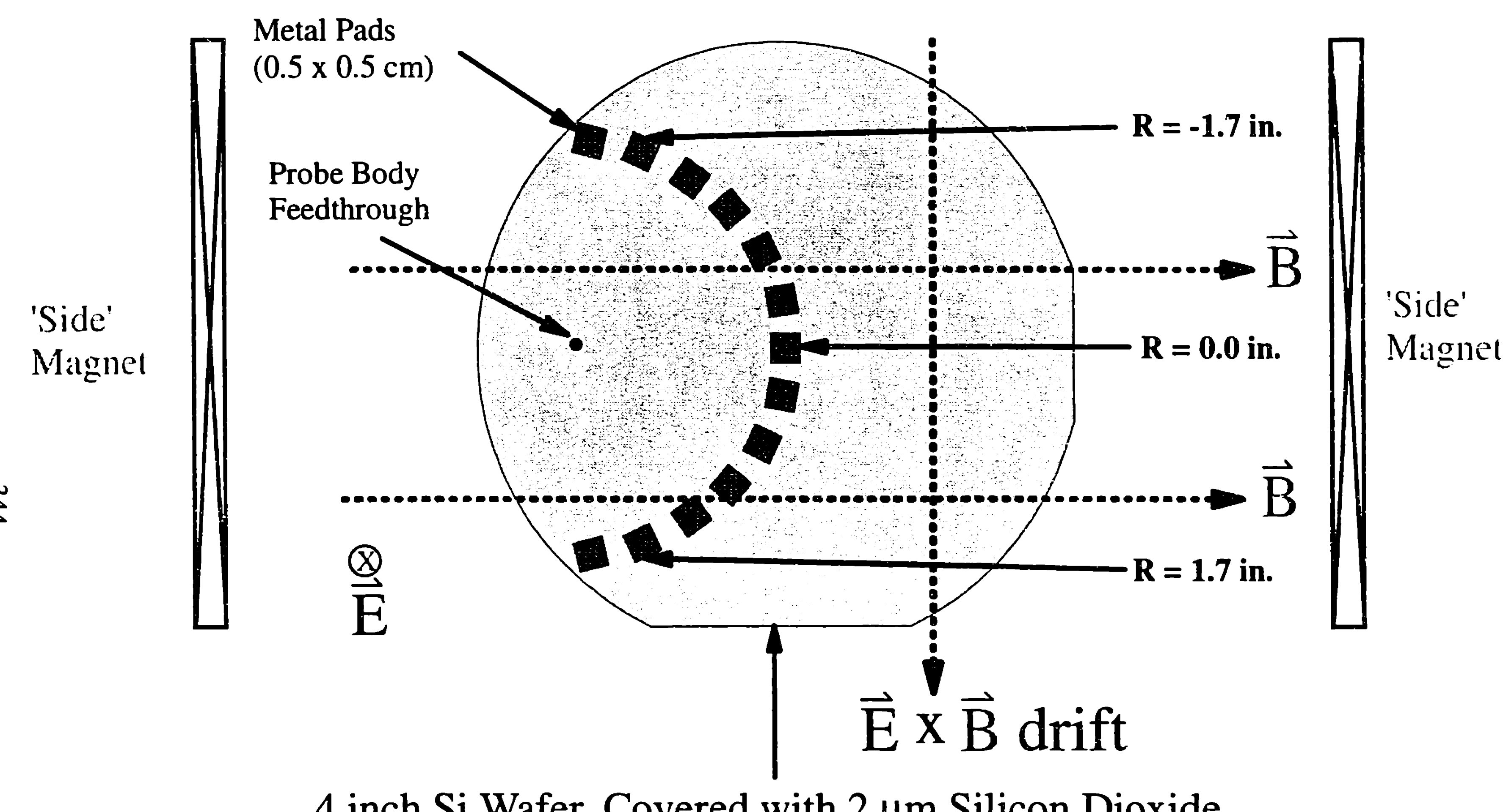


Figure 7-4: Test Wafer Design Used to Measure Surface Potential Variations. The Magnets Shown as 'Side' Magnets were Turned on to Produce a Magnetic Field in the Direction Indicated.

path shown by the metal pads in Figure 7-4. The coupling of metal pads to the wafer did influence the data. Repeatable results were obtained by using a very thin layer of silver paint to bind the metal pads to the wafer surface. The magnetic field inside the AME-5000 typically rotates during etching. For these experiments, a number of magnetic field configurations were used: (1) the magnetic field was held 'fixed' using the 'side' magnets whose position is shown in Figure 7-4; (2) the magnetic field was held 'fixed' using the 'front' magnets (located 90° from the 'side' magnets); and (3) the magnetic field was rotated at 0.5 or 1.0 Hz.

Experiments were conducted using 40 SCCM of Ar or Cl_2, at 100 mTorr, 250 W of rf power measured at the generator, with a cathode temperature of 20°C and with 4 Torr of He backside cooling.

7.3 Results

7.3.1 Surface Potentials

Surface potentials were measured on the metal pads initially without a magnetic field. The results are shown in Figure 7-5, which is a plot of the measured DC voltage on a pad (DC bias) versus the pad position (or pad number) on the wafer; the average of several measurements (triangular symbol) is plotted, along with error bars and a best fit line. In the absence of a magnetic field, the DC bias is fairly uniform across the wafer. The measurable variation in the surface potential is within the experimental repeatability.

However, a much different situation is seen with a magnetic field, as is shown in

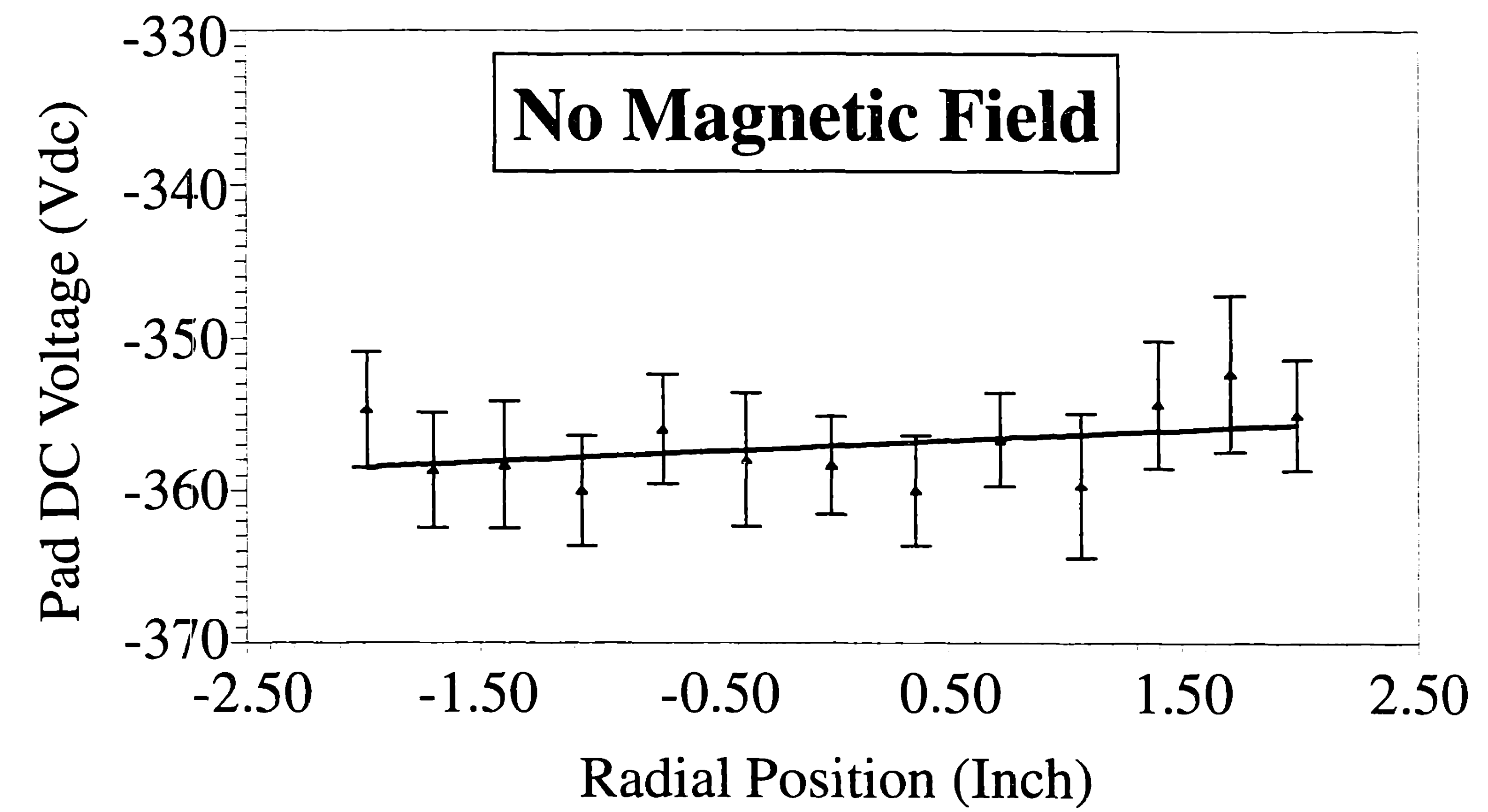


Figure 7-5: Wafer Surface Potential Variations as a Function of Position Without a Magnetic Field. In This Case, Essentially No Variation is Seen. Average, Error Bars and the Best Fit Line Are Shown.

Figure 7-6, which is for a 100 Gauss field strength, with the magnetic field held fixed using the 'side' magnets. In this case, there is a significant potential variation across the wafer (in the direction perpendicular to the magnetic field), as shown by the solid line. This potential variation follows the magnetic field direction as shown by the dashed line, which is for a 100 Gauss magnetic field with the magnetic direction reversed 180 degrees from that of the solid line. The magnetic field decreases the DC bias from about -360V to about -265V, which is consistent with other observations [Kuypers and Hopman, 1990]. No variations in the potential were observed in the direction parallel to the magnetic field. The observed potential difference is consistent with the charging current distributions across a wafer measured using MOS capacitors in the AME-5000 [Shin *et al.*, 1993b]. Rotating the magnetic field during etching will subject devices towards the wafer edge to greater potential variations. This effect has been seen experimentally as a difference in the center to edge oxide charging current [Hills, 1993].

Despite the difference in operating conditions and reactor, the variation in the DC bias across the wafer is of the same order as that measured by Tsuzuki *et al.* [1993] who observed a difference of 30 V for a 67.5 mT, 26 W Ar discharge with a field strength of 400 Gauss; they observed no potential variations parallel to the magnetic field, or without a magnetic field.

7.3.2 Plasma Potentials

The plasma potential was measured in addition to the wafer surface potential. Figure 7-7 shows the measured plasma potential variations as a function of height above

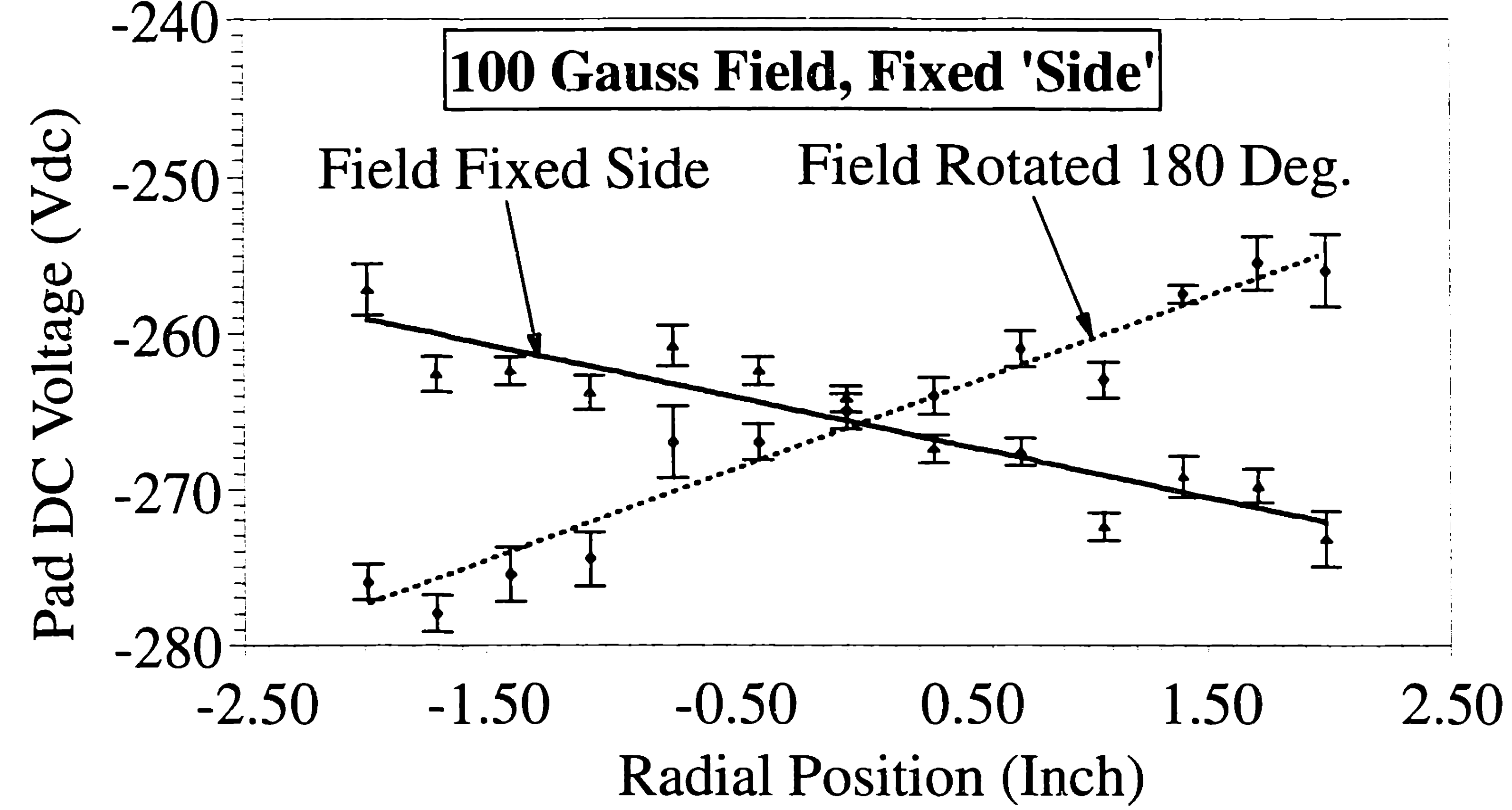


Figure 7-6: Wafer Surface Potential Variations as a Function of Position With a 100 Gauss Magnetic Field, Fixed 'Side'. Spatially Varying Potential Variations are Observed; these Variations Follow the Field Orientation. Measurements are Perpendicular to the Magnetic Field Direction. Average, Error Bars and the Best Fit Line Are Shown.

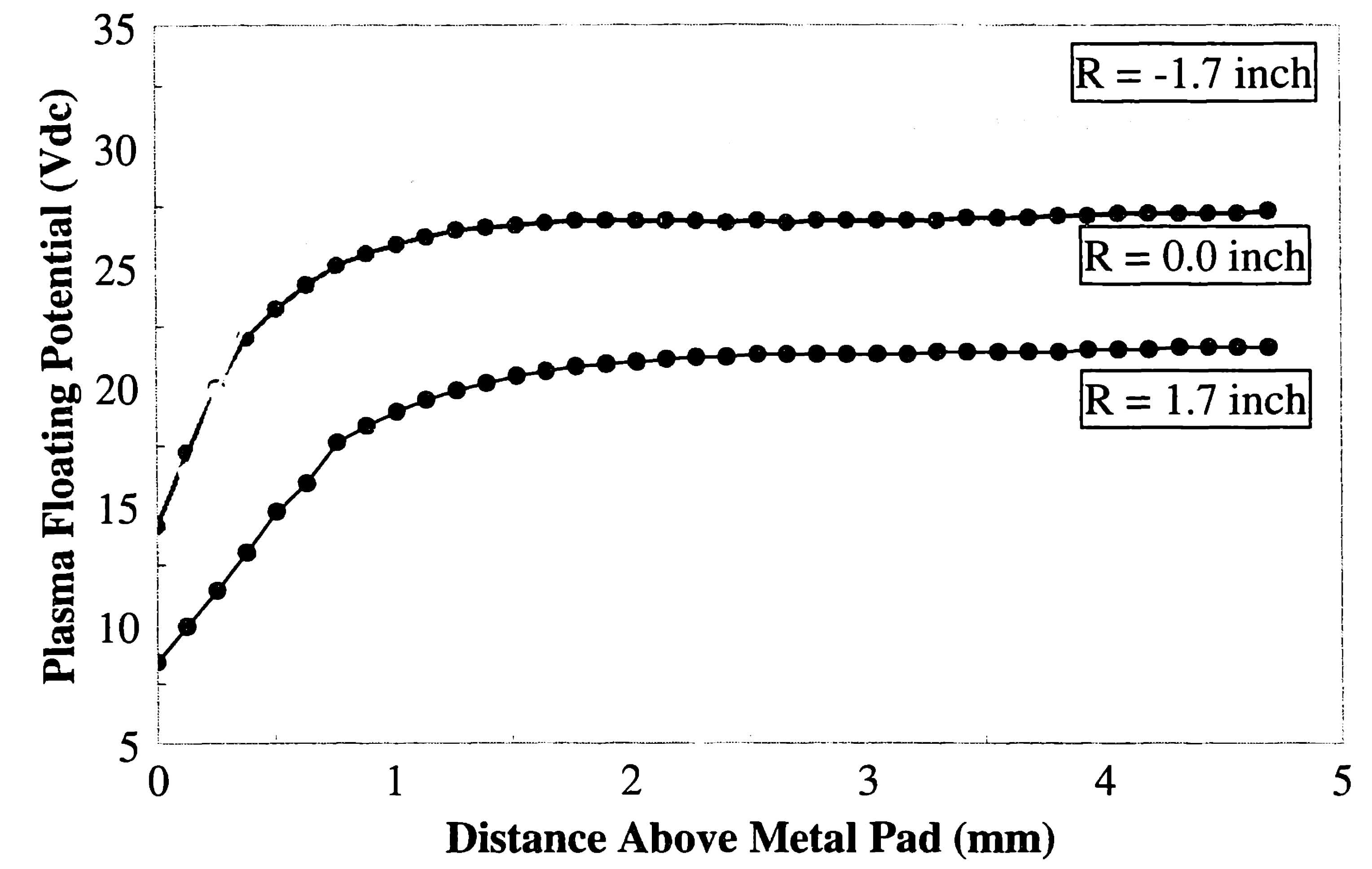


Figure 7-7: Plasma Floating Potential as a Function of Axial Position Above Three Metal Pads (Marked on Figure 7-4 as R = 0.0, ±1.7 in.) The Sheath Width and Plasma Floating Potential Vary as a Function of Position (B = 100 Gaus, Fixed 'Side').

3 of the metal pads on the wafer surface for a 100 Gauss magnetic field. The location of these pads (R = 0, ±1.7 in.) is indicated on Figure 7-4. The sheath thickness varies from around 1 mm to around 1.5 mm across the wafer. The plasma floating potential varies by about 10 volts across the wafer, which is about one-half the variation measured on the wafer surface. The electric field strength calculated from the initial slope of this plot yields field strengths ranging from 300 V/cm at R = -1.7 in. to 100 V/cm at R = 1.7 in, with bulk fields ranging from 5 to 2 V/cm for the 2 points, respectively. The plasma floating potential as a function of radial position is plotted in Figure 7-8 for scans taken at heights of 1.02, 1.52, 2.54, 3.56 and 4.06 mm above the metal pads. Again, we see the same behavior as in Figure 7-6. The difference in scans at 1.02 and 1.52 mm indicate that the plasma potential is still varying with axial position. The scans at 2.54, 3.56 and 4.06 mm were taken in the plasma bulk as evidenced by the lack of variation in the floating potential with axial position.

Similar variations in the plasma potential have been observed by others. Fang and McVittie [1992] observed 15 V variations perpendicular to the magnetic field and essentially no variations parallel to the field, or in the absence of a field for a 100 mT O_2 discharge with an 80 Gauss magnetic field. Measurements of the rf plasma voltage using a capacitive probe in the same type of reactor for a N_2 discharge with a 50 Gauss field (power and pressure unknown) yielded similar results [Savas and Donohoe, 1989]. The rf plasma voltage varied perpendicular to the magnetic field by about 120 V while variations parallel to the field and without a field were on the order of 20 volts. Field strengths of 200 to 3000 V/cm in the sheath and 4 V/cm in the bulk were reported. However, their axial spatial resolution was only 0.5 mm while the micrometer on out

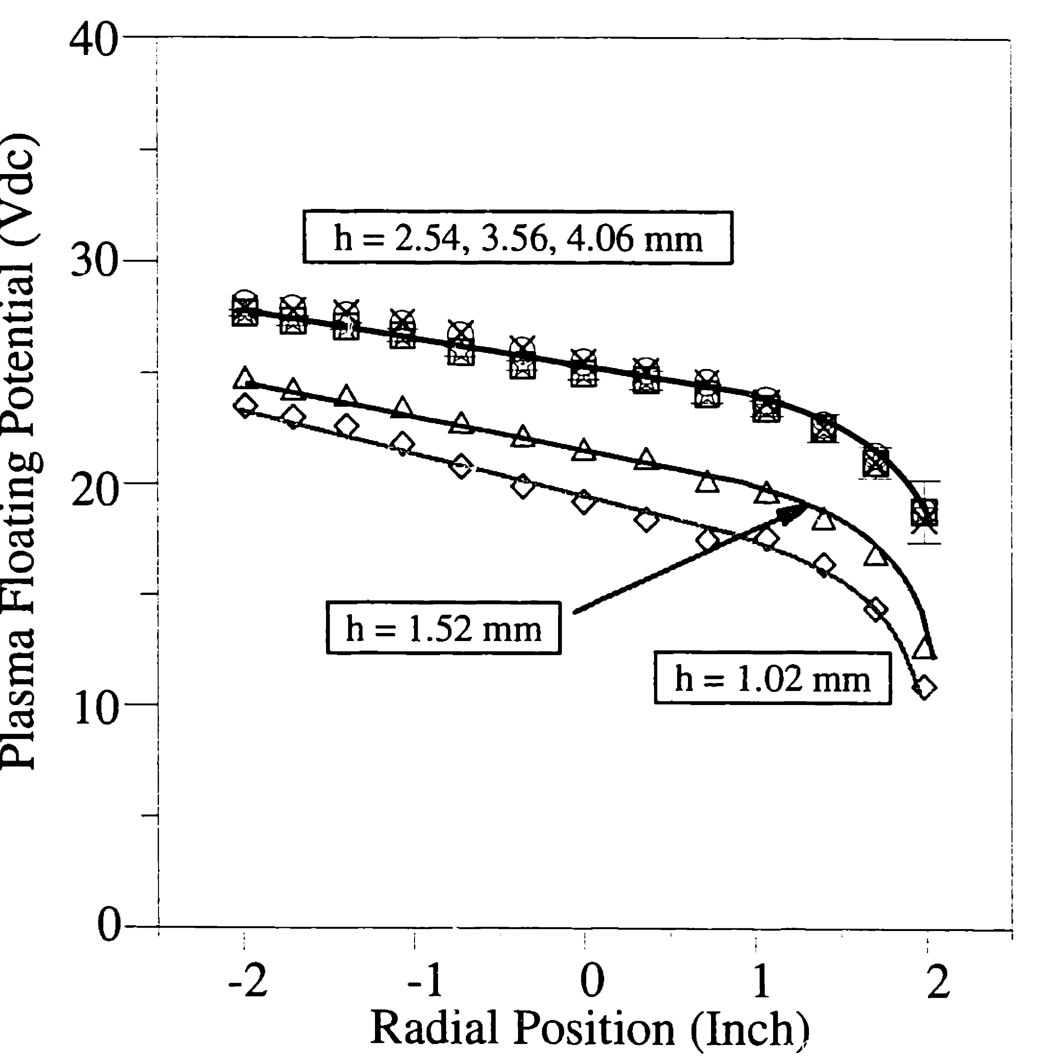


Figure 7-8: Plasma Floating Potential as a Function of Position over the Metal Pads used to Measure Surface Potential. Scans at Varying Height Above the Metal Pads.

feedthrough could be read down to 0.025 mm; steps on the order of 0.13 mm were used during data collection.

Plasma floating potentials were measured as a function of height above the wafer center for Cl_2 discharges (Figure 7-9) in addition to Ar discharges (Figure 7-7); the data for Ar is also shown on the plot. Scans for Cl_2 with magnetic fields of 0 and 75 Gauss allow us to estimate the sheath thickness as 3.6 and 2.2 mm, respectively.

The effect of a rotating magnetic field is shown in Figure 7-10, which is a plot of the plasma floating potential as a function of time above the wafer center for a Cl_2 discharge with a magnetic field of 75 Gauss. The field rotation frequency was 1.0 Hz (black line) and 0.5 Hz (gray line). The field rotation was accomplished using a 36 segment approximation to circular rotation for the current to the 4 magnets (2 'side' and 2 'front'). The potential at the center varies periodically as the magnetic field rotates. This variation is not smooth, but is rather jagged; the potential changes by a magnitude of 20 V during the field rotation.

Potential variations in the direction perpendicular to the magnetic field were shown in Figures 7-6 and 7-7. The magnetic field was rotated 90 degrees by using the 'front' magnets instead of the 'side' magnets. A scan at 0.52 inches above the wafer is shown in Figure 7-11 for a Cl_2 discharge at 100 Gauss. The direction of scanning is now parallel to the magnetic field instead of perpendicular to it. The floating potential profile is now peaked at the center instead of changing monotonically across the scan path. This is due to the fact the center of the wafer experiences the largest effect of the "E cross B" drift, while the edges of the wafer (±2 inches) experience the same effect of the drift.

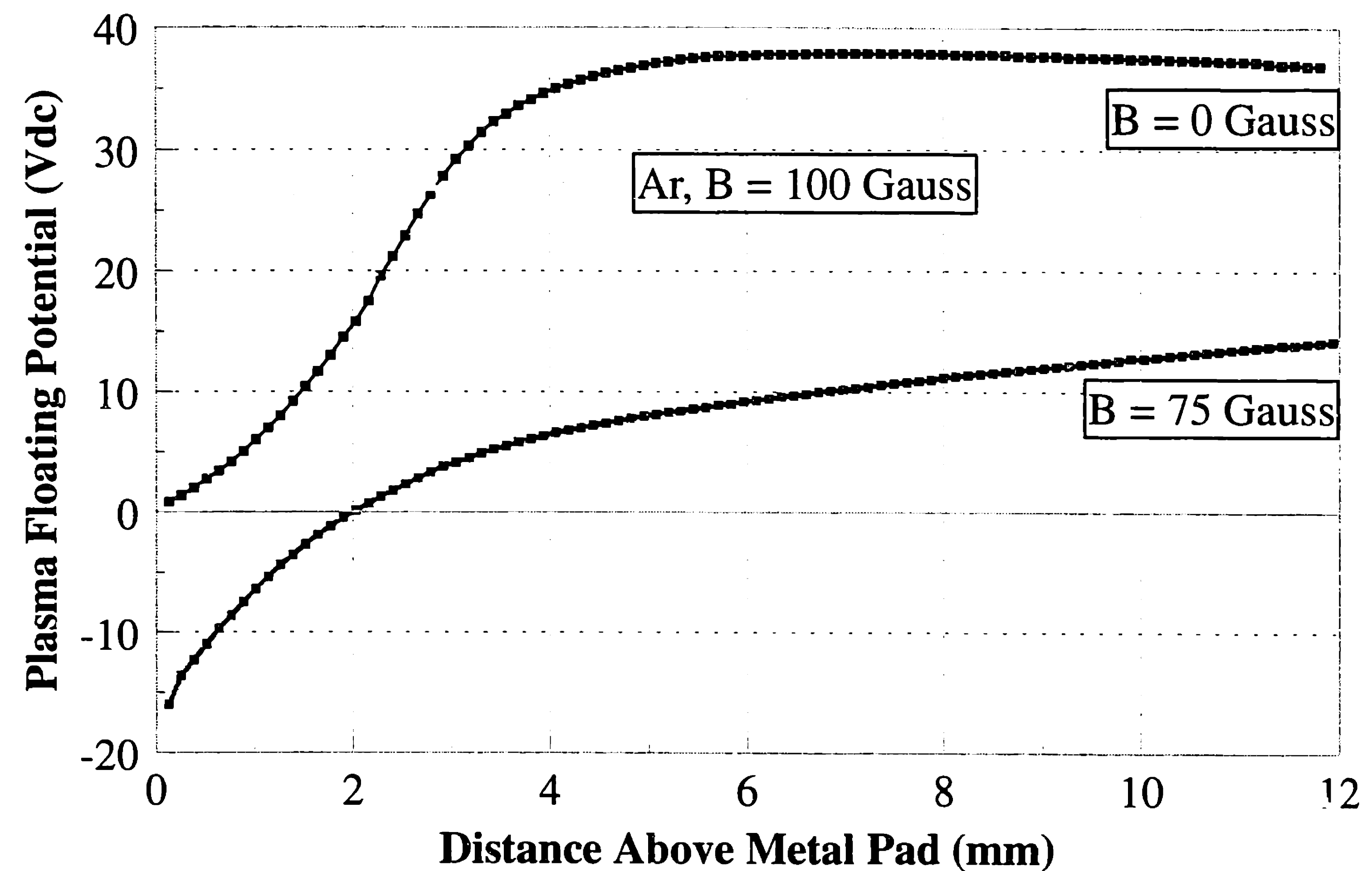


Figure 7-9: Plasma Floating Potential as a Function of Height at Wafer Center for a Cl_2 Discharge With Magnetic Fields of 0 and 75 Gauss. Also Plotted is Data of Figure 7-7 for Ar at B=100 Gauss at the Wafer Center

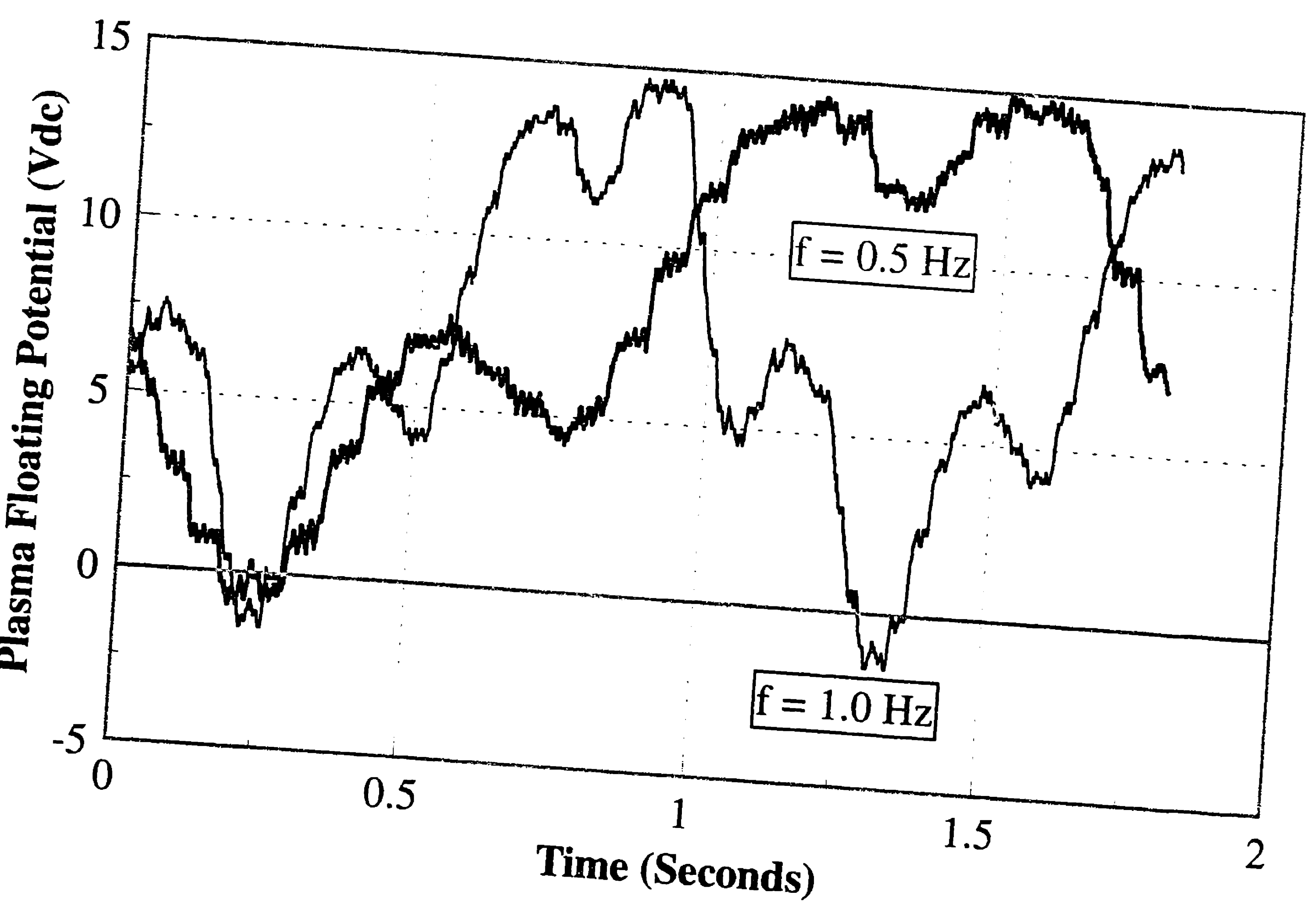


Figure 7-10: Plasma Floating Potential Above Center of Wafer in Cl2 Discharge With 75 Gauss Magnetic Field Rotating at 0.5 and 1.0 Hz

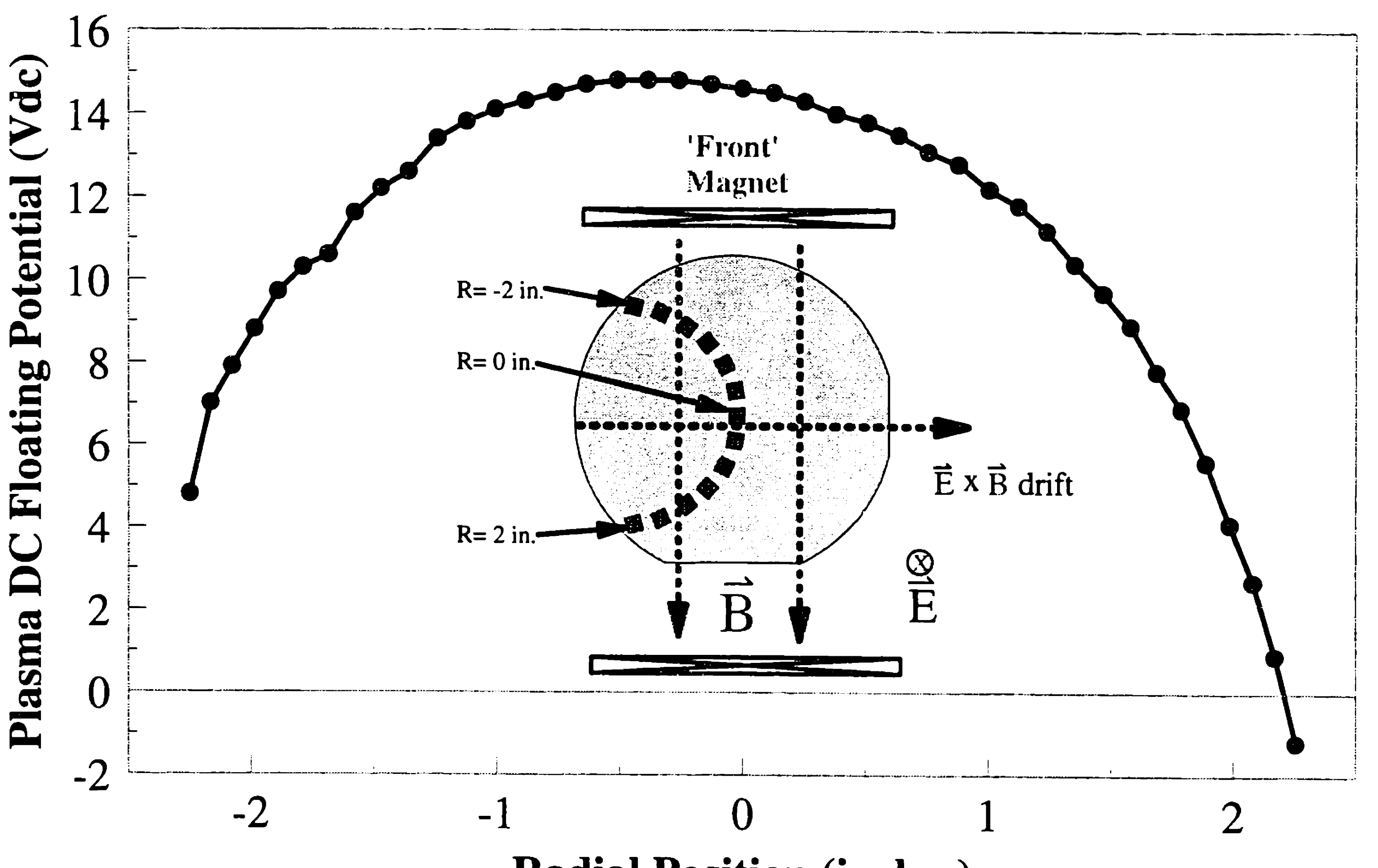


Figure 7-11: Plasma Floating Potential as a Function of Position Across Wafer, 0.52 inches Above Wafer. Cl_2 Discharge at 100 G. Note: Field is Fixed 'Front' and Not 'Side' as is Shown With the Wafer Image.

7.3.3 Process Variables

Process variables affect both the DC bias and the potential difference across a wafer. The effects of pressure and magnetic field on DC bias at the wafer center for a 250 W Ar discharge are shown in Figure 7-12 for pressures between 0 and 250 mTorr and magnetic fields between 0 and 125 Gauss. In all cases, increasing the magnetic field decreases the DC bias. Similarly, decreasing the pressure increases the DC bias. However, the effect of pressure decreases as magnetic field is increased.

It is desirable to minimize wafer surface potential variations in order to minimize the possibility of gate oxide damage. To study the effect of process variables on potential variations, a neural network experimental design was constructed using NNAPER [NNAPER, 1993]. The variables and ranges examined are shown in Table 7-2.

Table 7-2

Designed Experiment Input Parameters and Ranges

Process Variable	Minimum	Maximum	Units
Power	100	400	Watts
Pressure	10	250	mTorr
Magnetic Field	0	120	Gauss

The measured outputs for the experimental design were the maximum surface potential variation across the wafer, and the value of the DC bias at the wafer center. The experimental design consisted of 24 experiments, 20 points of training data for the

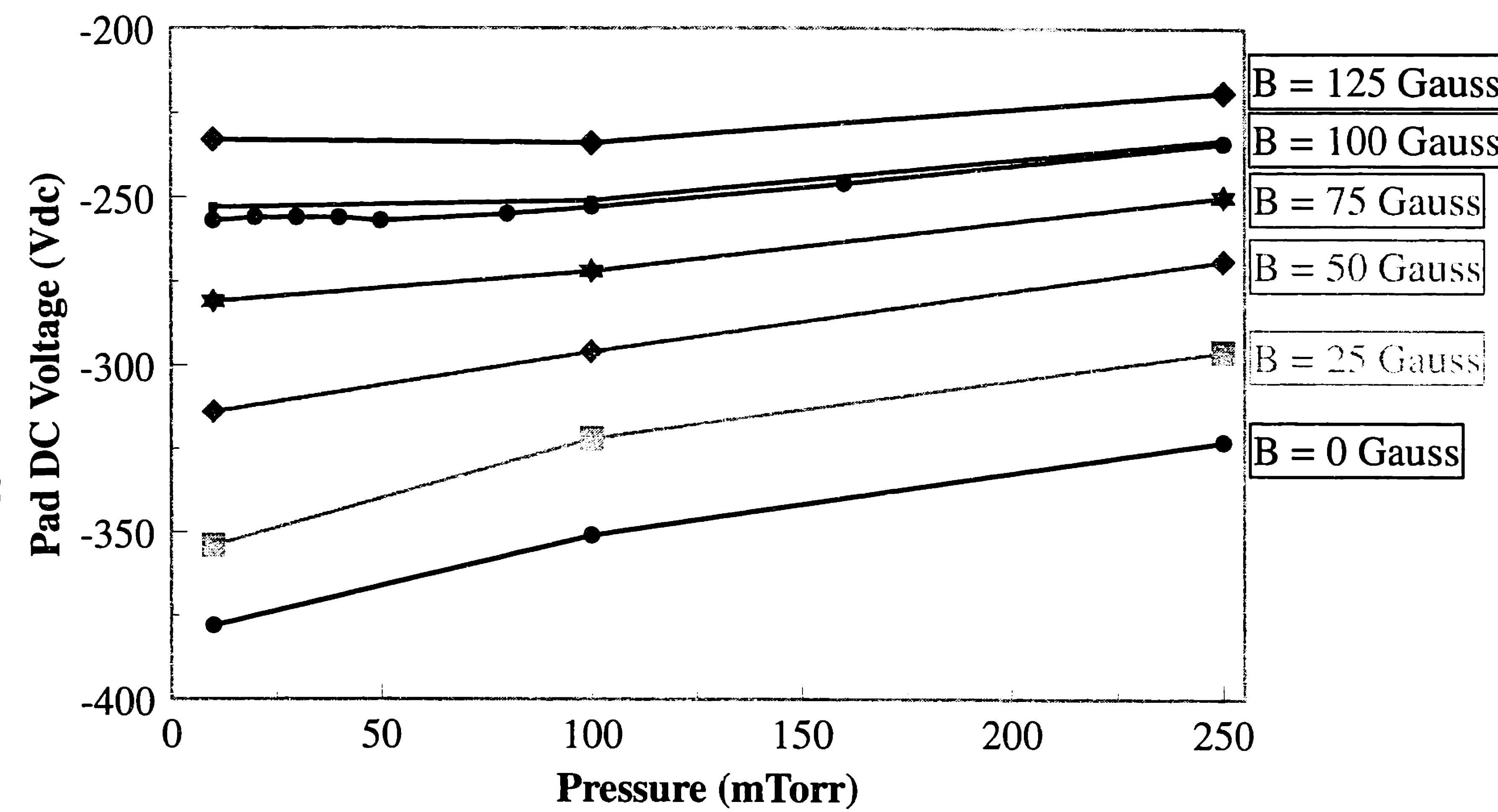


Figure 7-12: Affect of Pressure and Magnetic Field on Measured Pad Voltage (DC Bias) at Wafer Center in Ar Discharge.

network and 4 test points. In addition to the minimum 24 experiments, 25 other experiments were performed for a total of 41 training points and 8 test points. The process conditions, point type ("S" = Simplex, "F" = Factorial, "C" = Center, "T" = Test) and the measurements are listed in Table 7-3.

Some of the trends observed in the data are shown in Figures 7-13, 7-14, and 7-15, which are plots of the modeled potential differences (ΔV) as a function of power and pressure for magnetic field strengths of 0, 60 and 120 Gauss. At 0 Gauss (Figure 7-13), the potential difference is quite small; over a wide range of power and pressure, ΔV is less than 4 volts. Only at high power (400 W) and low pressure (10 mTorr) does it increase to around 5.5 V. The presence of a magnetic field strongly influences ΔV; a field of 60 Gauss increases the over all potential difference and enhances the difference at high power and low pressure (Figure 7-14). In this case, ΔV_{max} is on the order of 20 volts. Further increasing the magnetic field to 120 Gauss (Figure 7-15) tends to increase the range of power and pressure over which the maximum potential difference is observed. The trends can be summarized as follows: the potential difference on the wafer surface increases with decreasing pressure, increasing power, and increasing magnetic field strength.

To decrease potential variations on the wafer surface, pressure should be increased (so that $N_{mag,c}$ is increased), power decreased and magnetic field strength decreased. These trends conflict with other process trends. Lower pressure is desirable for profile control. Higher power and field strength are desirable for increased throughput. The effects of magnetic field and pressure on oxide charging current (and thus device damage) measured by others [Hills, 1993] agree with the predictions of our neural network model.

The effects of process conditions on the DC bias measured at the wafer center was

Table 7-3

List of Designed Experiments and Outputs

Mag. Fld. (Gauss)	Power (Watts)	Pressure (mTorr)	"Type"	|Δ V| (Volts DC)	DC Bias (Volts DC)
60	250	130	C	10	-300
60	250	130	C	10	-301
60	250	130	C	12	-303
0	250	130	S	2	-371
60	100	130	S	8	-186
120	250	130	S	18	-262
60	250	10	S	20	-317
60	400	130	S	18	-364
60	250	250	S	8	-260
27	125	177	T	6	-211
60	250	130	C	11	-285
25.4	163	199	F	6	-238
25.4	337	199	F	12	-358
117	201	196	T	9	-213
94.6	163	199	F	9	-203
25.4	163	61	F	9	-247
60	250	130	C	12	-271
62.8	363	102	T	13	-342
25.4	337	61	F	10	-395
94.6	337	199	F	16	-286
94.6	337	61	F	18	-314
39	251	54	T	6	-316
60	250	130	C	9	-282
94.6	163	61	F	14	-208
60	250	130	C	7	-284
60	250	130	R	12	-286
60	250	130	T	11	-288
0	250	130	R	8	-356
60	100	130	S	9	-176
120	250	130	S	17	-244
60	250	10	S	21	-296
60	400	130	S	15	-379
60	250	250	S	9	-274
27	125	177	T	6	-218
60	250	130	C	8	-291
25.4	163	199	F	6	-245
25.4	337	199	F	8	-363
117	201	196	T	12	-224
94.6	163	199	F	11	-213
25.4	163	61	F	10	-255
60	250	130	C	11	-287
62.8	363	102	T	14	-359
60	250	130	C	12	-305
60	250	130	C	11	-306
60	250	130	C	10	-306
0	250	130	S	1	-375
60	100	130	S	8	-186
120	250	130	S	21	-256
60	250	10	S	22	-321

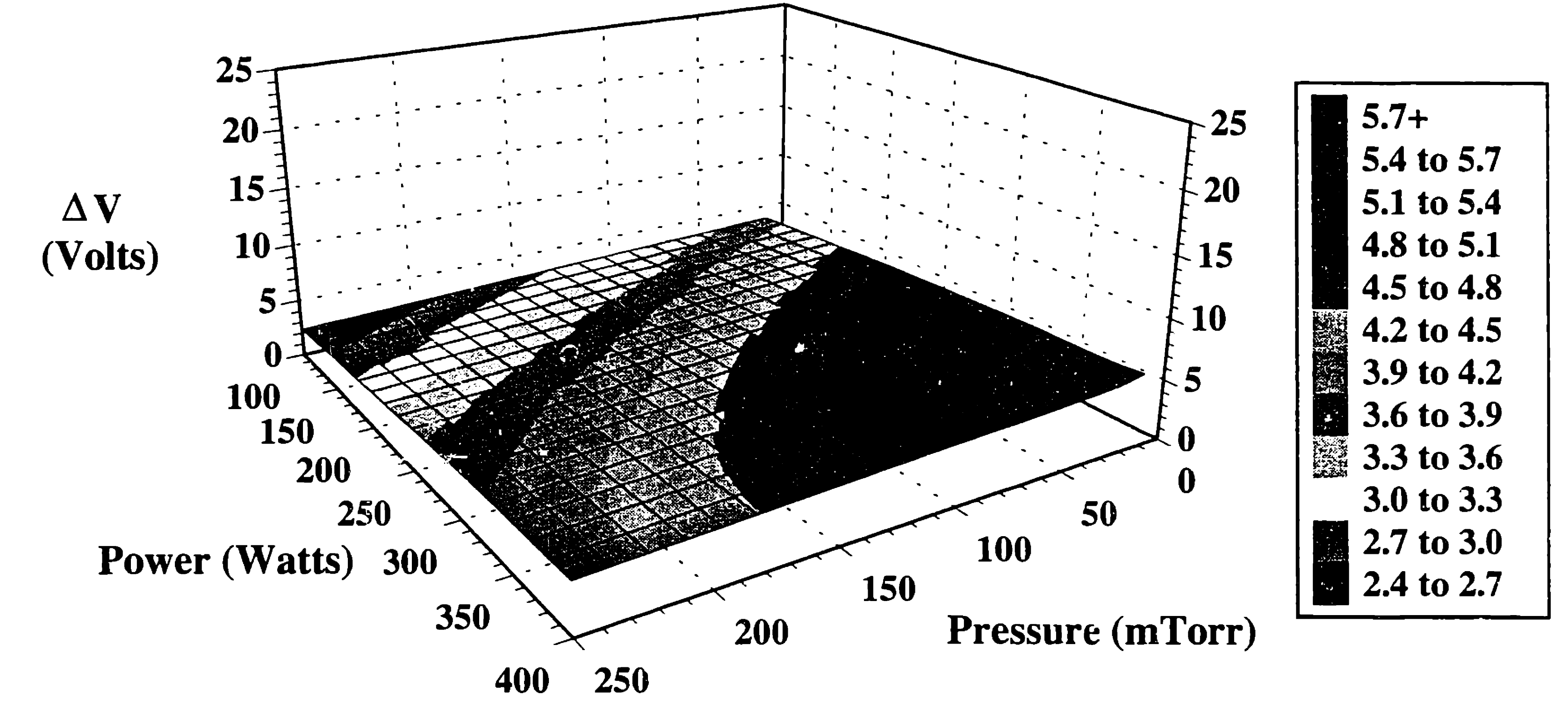


Figure 7-13: Surface Potential Variations With Process Variables. Wafer Surface Potential Plotted as a Function of Plasma Power and Pressure for a Magnetic Field Strength of 0 Gauss.

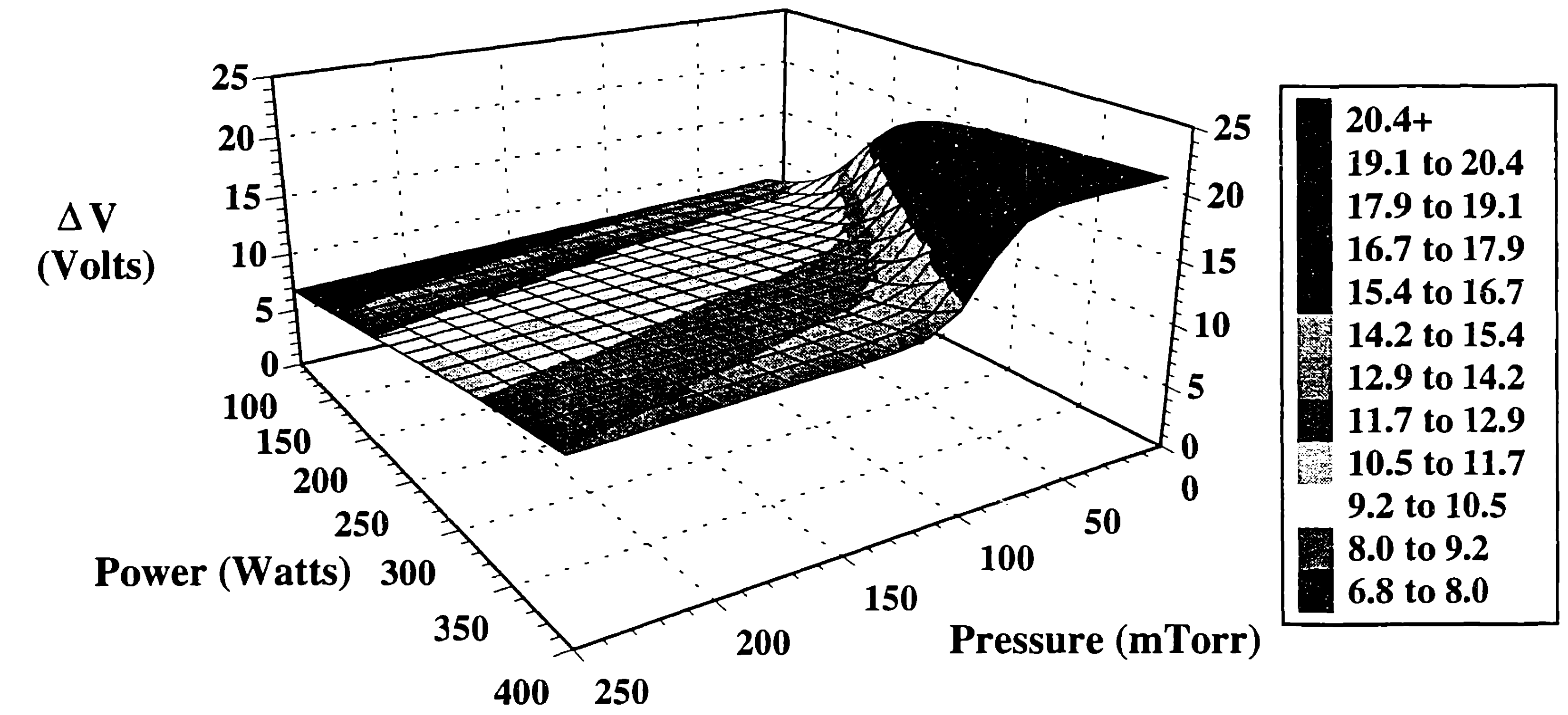


Figure 7-14: Surface Potential Variations With Process Variables. Wafer Surface Potential Plotted as a Function of Plasma Power and Pressure for a Magnetic Field Strength of 60 Gauss.

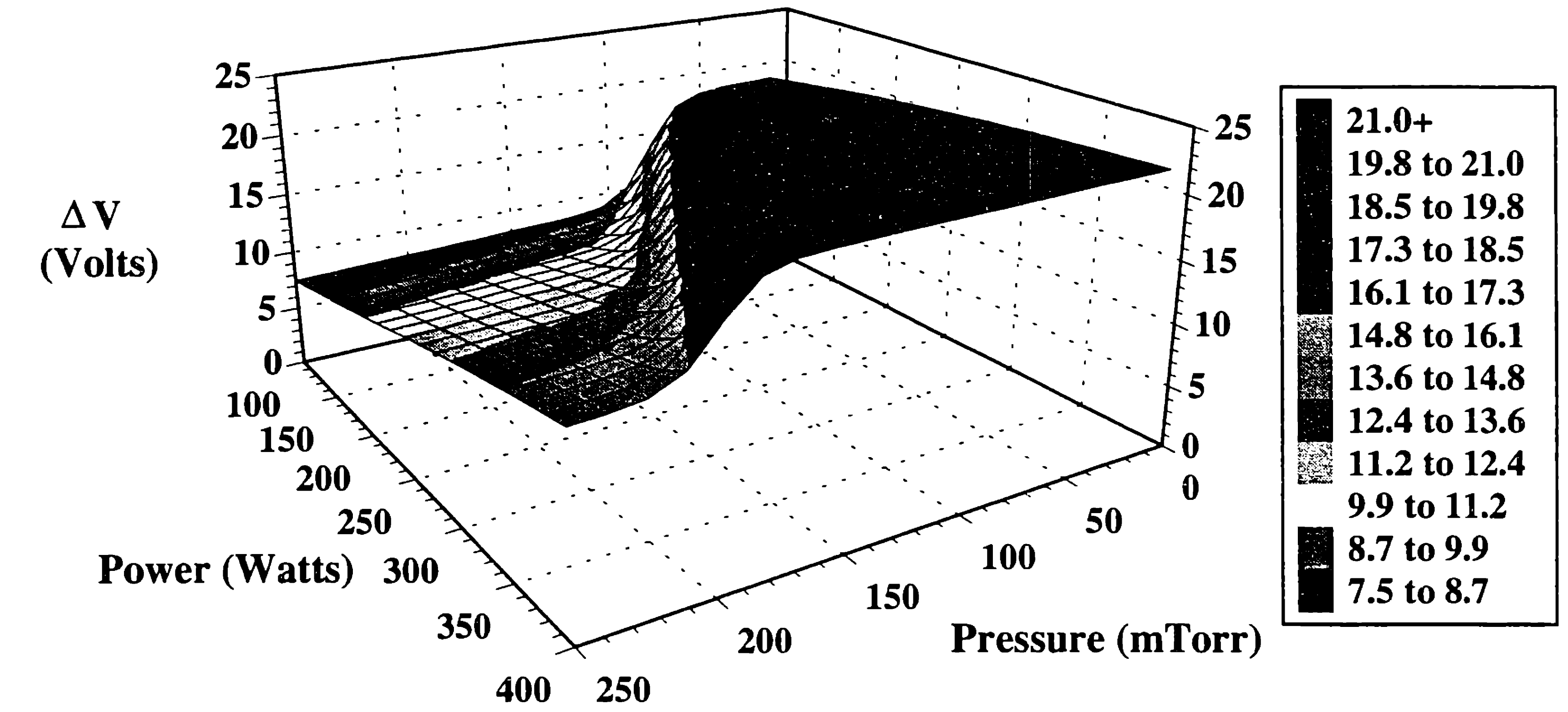


Figure 7-15: Surface Potential Variations With Process Variables. Wafer Surface Potential Plotted as a Function of Plasma Power and Pressure for a Magnetic Field Strength of 120 Gauss.

also studied; the neural network model results are shown as plots of the DC bias as a function of pressure and power for magnetic field strengths of 0, 60, and 120 Gauss in Figures 7-16, 7-17 and 7-18. The DC bias is a negative voltage; for that reason we will be referring to the magnitude of the DC bias, $|V_{dc}|$ in the following discussion. Increasing the magnetic field strength tends to decrease $|V_{dc}|$. Decreasing power strongly decreases $|V_{dc}|$. Increasing pressure slightly increases $|V_{dc}|$. Some plasma etchers allow for easy measurement of DC bias; some use it as a control variable. The polysilicon etching chamber on the AME-5000 does not allow for measurement of DC bias. These probe measurements are important to understanding how process conditions affect ion bombardment energy.

7.4 Conclusions

Wafer surface potentials have been measured in an Applied Materials Precision 5000 plasma etcher. In the absence of a magnetic field, no variation in the surface potential with position is observed. In the presence of a magnetic field, wafer surface potential variations are observed in the direction perpendicular to the magnetic field, on the order of 10 to 20 volts for a 100 Gauss magnetic field. The variations are caused by plasma nonuniformities generated by the magnetic field. Similar variations are seen in the plasma floating potential. The magnitude of the variations increased with decreasing pressure, increasing power and increasing magnetic field strength.

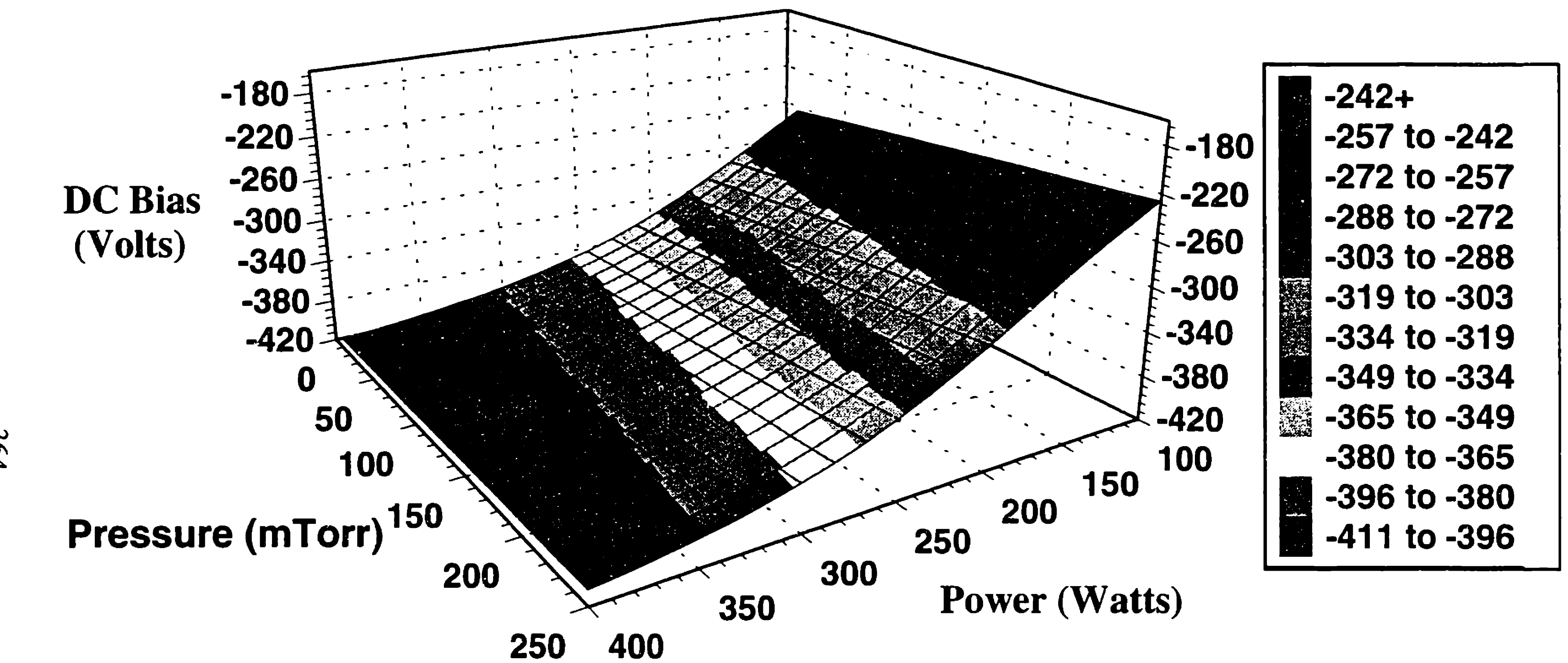


Figure 7-16: DC Bias Variations With Process Variables. DC Bias Plotted as a Function of Plasma Power and Pressure for a Magnetic Field Strength of 0 Gauss.

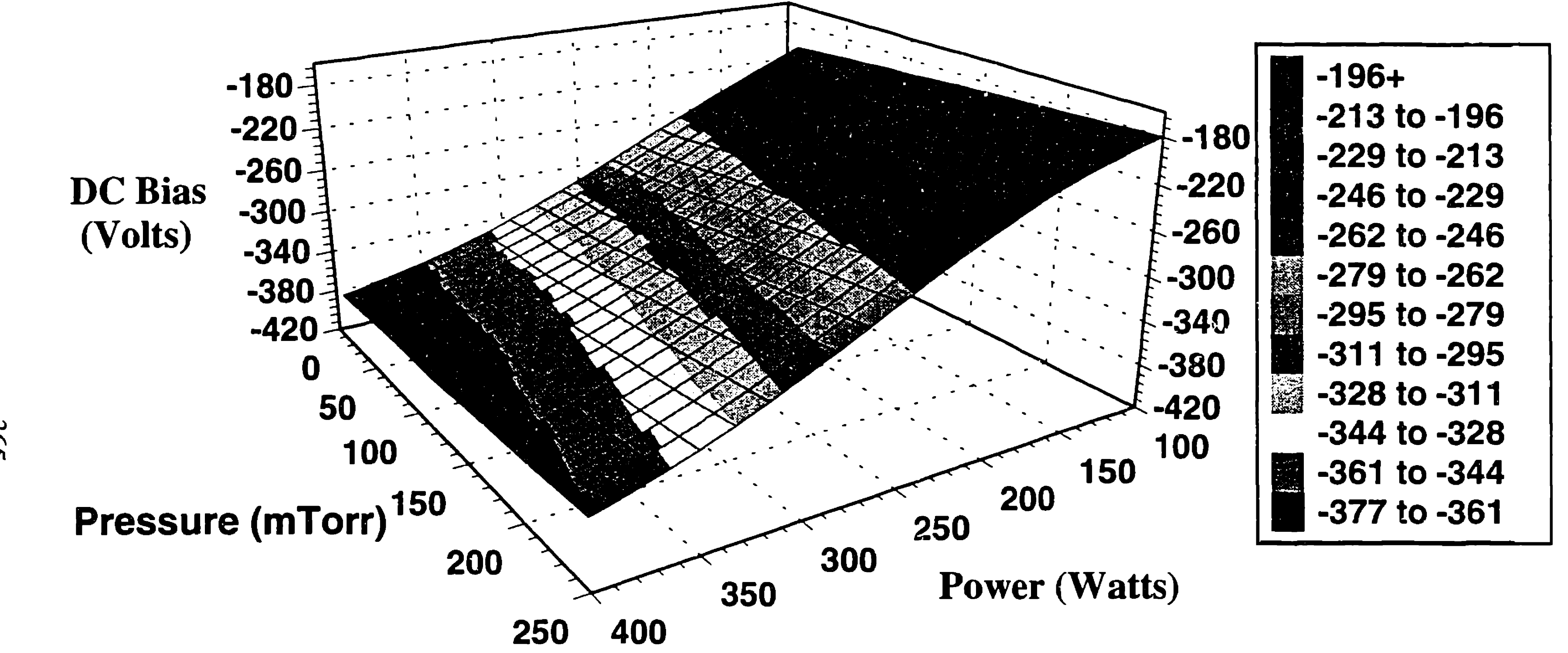


Figure 7-17: DC Bias Variations With Process Variables. DC Bias Plotted as a Function of Plasma Power and Pressure for a Magnetic Field Strength of 60 Gauss.

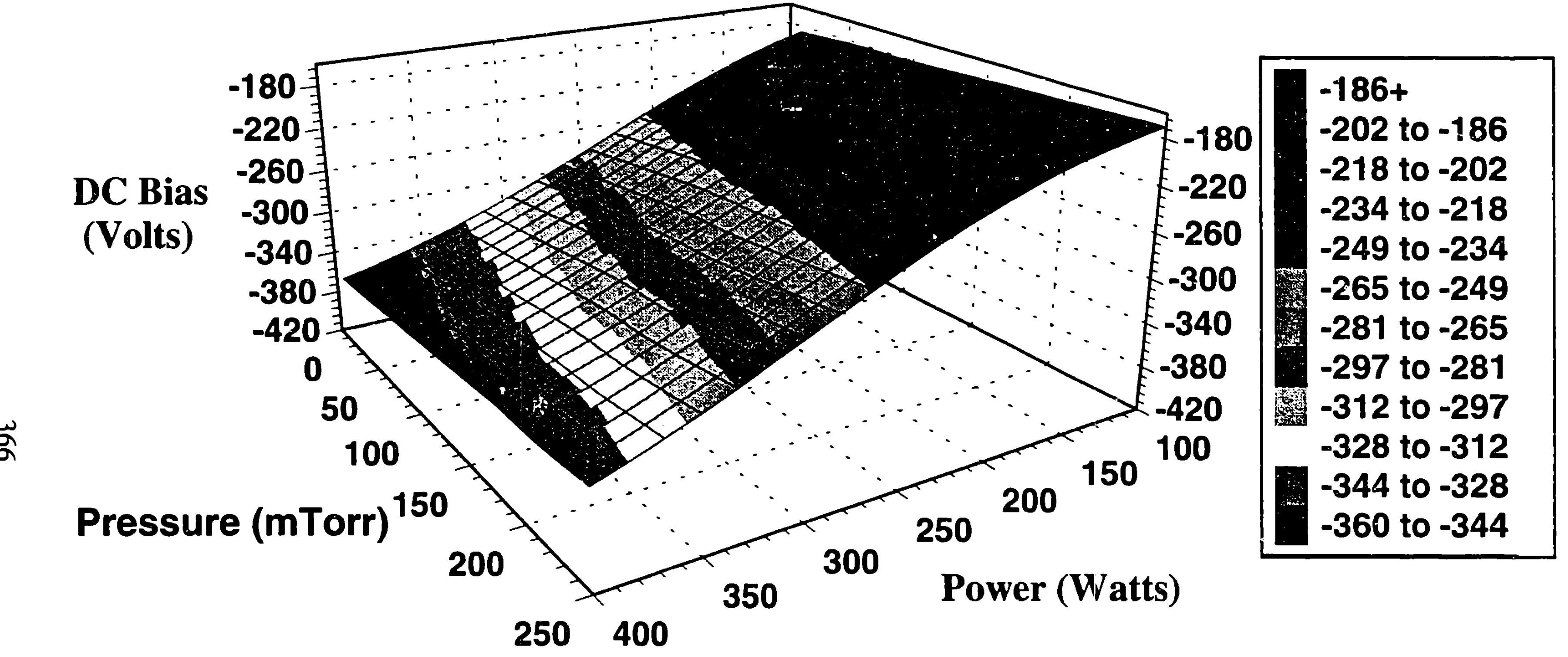


Figure 7-18: DC Bias Variations With Process Variables. DC Bias Plotted as a Function of Plasma Power and Pressure for a Magnetic Field Strength of 120 Gauss.

Chapter 8

Aspect Ratio Dependent Etching

Spatially resolved optical emission interferometry was measured for patterned polysilicon etching using a CCD camera. The test mask used contained gratings of various dimensions allowing us to analyze the phenomena of "Aspect Ratio Dependent Etching" (ARDE), *i.e.* the variation of etching rate with feature aspect ratio. With the use of the CCD system, the variation of a few percent in etching rate with aspect ratio can be observed. The aspect ratio dependence is a function of residence time (gas flowrate), pressure, magnetic field strength, temperature, feed gas composition, and the seasoning/history of the etcher. The following chapter describes the use of a new technique to quantify ARDE on a commercial etching reactor.

8.1 Introduction to ARDE

Aspect ratio dependent etching (ARDE), "RIE Lag" or "Pattern-Factor Dependent Etching" is the generalization that trenches (or holes) with a smaller spacing (diameter) etch slower than features with larger spacing (diameter) [Singh *et al.*, 1989]. This phenomena has been observed in a wide range of plasma chemistries for many different types of films. These include SiO_2 with CF_4/O_2 and CF_4/He [Singh *et al.*, 1989], HPR 204 (a novolac-based photoresist) with O_2 [Janes and Pilz, 1993], Si with CF_3Br/SF_6 [Lee and Zhou, 1991], Si with CCl_2F_2 [Chin *et al.*, 1985], n^+ poly-Si with Cl_2 and HCl

[Fujiwara *et al.*, 1990], poly-Si with Cl_2 [Fujiwara *et al.*, 1989], and GaAs with $Cl_2/SiCl_4$ [Cooper *et al.*, 1987]. A thorough review article has recently been published on ARDE [Gottscho *et al.*, 1992]. Chin *et al.* [1985] observed that the etching rate depended solely on the aspect ratio (AR, defined as the ratio of the etch depth to the line spacing) of the features, independent of the opening size, and decreased almost linearly with increasing aspect ratio.

There are a number of possible causes for ARDE. As was discussed in sections 1.5.1 and 1.5.4, the feature size (aspect ratio) may limit the arrival of angularly dependent fluxes (both ion and neutral). ARDE is often attributed to such 'shadowing' effects. Other possible causes include angularly dependent deposition and redeposition processes.

8.2 Method of Approach

In the past, ARDE has been measured by partially etching a film, and then measuring the etch depth on a cleaved wafer using an SEM or XTEM. However, this process is very slow and difficult to use for quantification of small etching differences. We used the newly developed technique of Full Wafer Interferometry (FWI) to measure etching rate variations as a function of aspect ratio and process conditions *in situ*.

8.2.1 CCD Interferometry

Spatially resolved etching rates were measured *in situ* during etching using FWI. Since this technique has been described in detail previously (Chapter 5), only a brief

description will be given here. Plasma optical emission (OE) is used as the illumination source for interferometry. Light is collected by a lens, passed through an optical band pass filter (in this case, a filter centered at 753.4 nm with a 32.0 nm FWHM), and is imaged onto a CCD array. The intensity as a function of time is analyzed at various locations on the image to determine the etching rates at these different positions.

Previously, a 16-mm focal length lens was used to view an entire 10-cm diameter wafer during etching. In this case, a pixel on the CCD corresponded to approximately 170 μm x 400 μm on the wafer surface. This region is rectangular because the pixels on the CCD camera used were not square. In this case, all of the dies on the wafer are visible (Figure 8-1). However, detailed analysis of the etching behavior for different regions on the test mask can not be made at this resolution.

A 50-mm focal length lens was used to increase the image magnification; this resulted in increase spatial resolution at the expense of a smaller field of view. Using this lens, a pixel on the CCD corresponded to about 30 μm x 70 μm on the wafer surface, with a total field of view of around 4 cm^2. A typical view under these conditions is shown in Figure 8-2, which shows a view of the four dies visible at the wafer center during an etch. The length of a side of a die is 1.0-cm and is shown with the double-ended arrow on this figure.

8.2.2 Test Mask

A grating pattern was chosen for analysis because it consists of the same pattern repeated over a large area (approximately 0.1 mm^2) making it easy to visually identify the

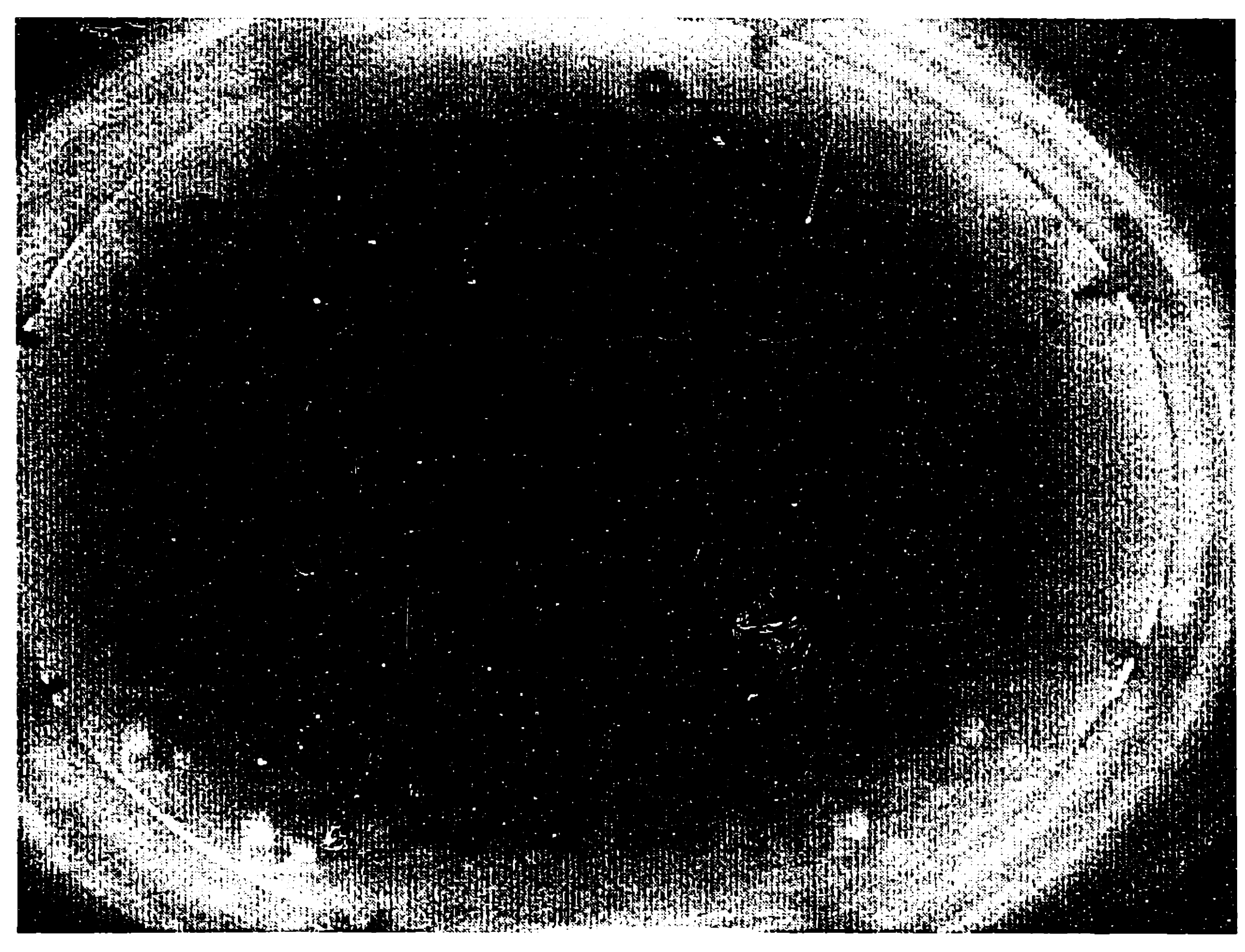


Figure 8-1: View of Patterned Wafer Taken With f=16-mm Lens.

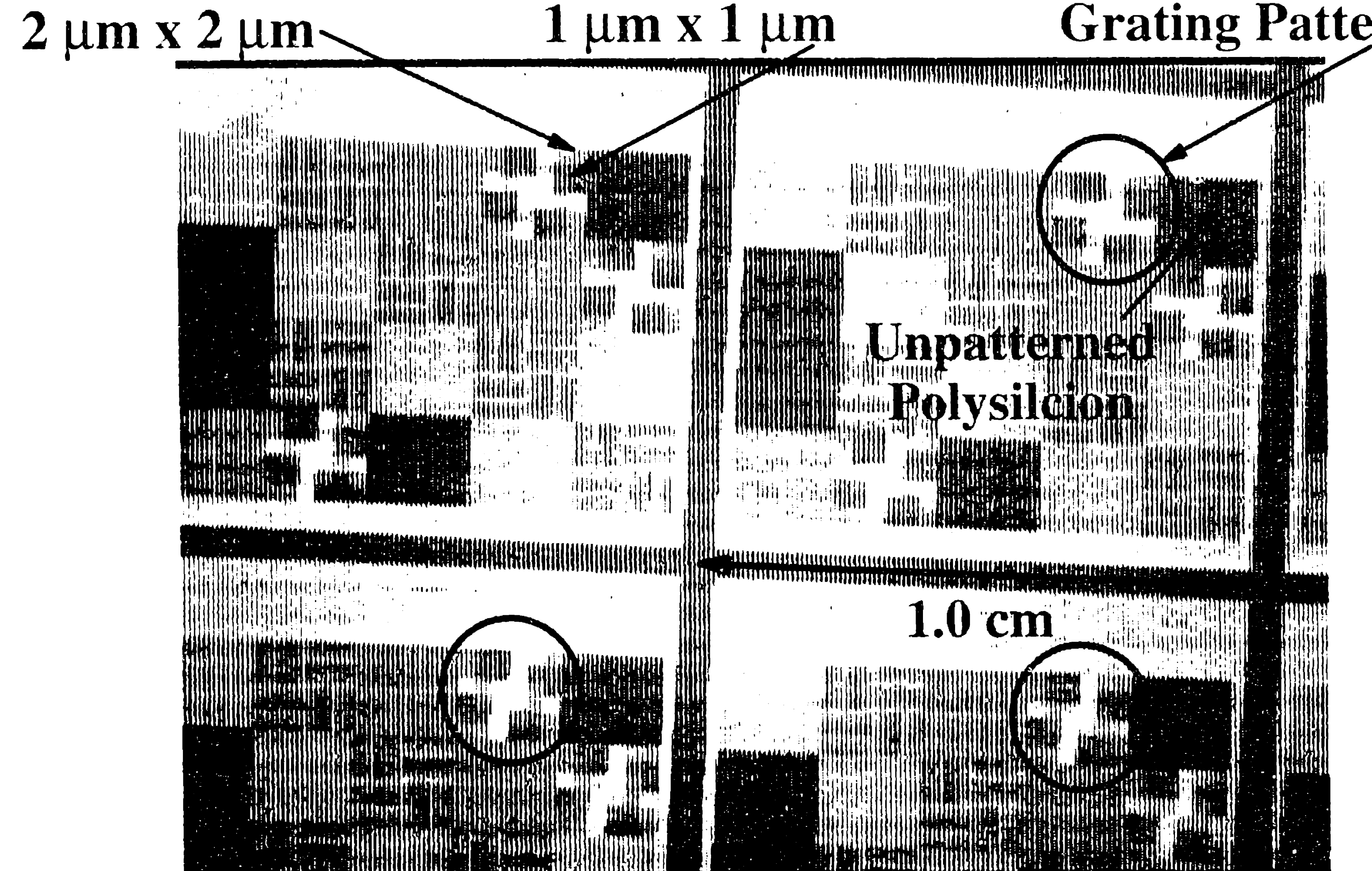


Figure 8-2: View of Wafer Center During Etch, with the Region of the Die used in the ARDE Analysis Indicated, Along with the Location of the Patterns in that Region.

different portions of the pattern. The diffraction grating pattern used in this analysis is shown circled on Figure 8-2. This pattern contains 9 different sets of line width and line spacing combinations: widths of 0.5, 1.0 and 2.0 μm and spacings of 0.5, 1.0 and 2.0 μm. Unfortunately, the 0.5 μm lines and spaces did not print because the stepper used could not print features below 1 μm. The particular patterns of interest in this case are the 1 μm lines x 1 μm spaces and the 2 μm lines x 2 μm spaces, some of which are indicated.

A schematic representation of this analysis regions is shown in Figure 8-3. The four dies visible on Figure 8-2 are shown schematically in Figure 8-3A. A detailed view of the test mask is shown in Figure 8-3B. Each die (1-cm x 1-cm) region is composed of 20 "subdie". One of these subdie is shown in Figure 8-3C. The pattern on the "subdie" is generated by taking a basic pattern and rotating it about the center of the "subdie". Thus, the "subdie" contains 4 of the 1 μm x 1 μm regions of interest and 4 of the 2 μm x 2 μm regions of interest. The diffraction grating pattern is repeated three times on each die; in our analysis, only the top-most set of gratings is used on each die. Data from 4 regions on the indicated subdie, on 4 die visible at the center of the wafer was used in our measurements. Thus, the measurement for etching rate at each aspect ratio was calculated based upon 16 data points.

The wafers used in this analysis consisted of a 5000 Å polysilicon film grown over a 1000 Å silicon dioxide film and were patterned using about 10000 Å of KTI-820 photoresist. Given these film thicknesses, the patterns observed corresponded to a nominal aspect ratio of: AR=0 for the unmasked polysilicon; AR = 0.75 for the 2 μm lines x 2μm spaces; and AR=1.5 for the 1 μm lines x 1 μm spaces. Etching was performed on an Applied Materials Precision 5000 plasma etcher (AME-5000) in a

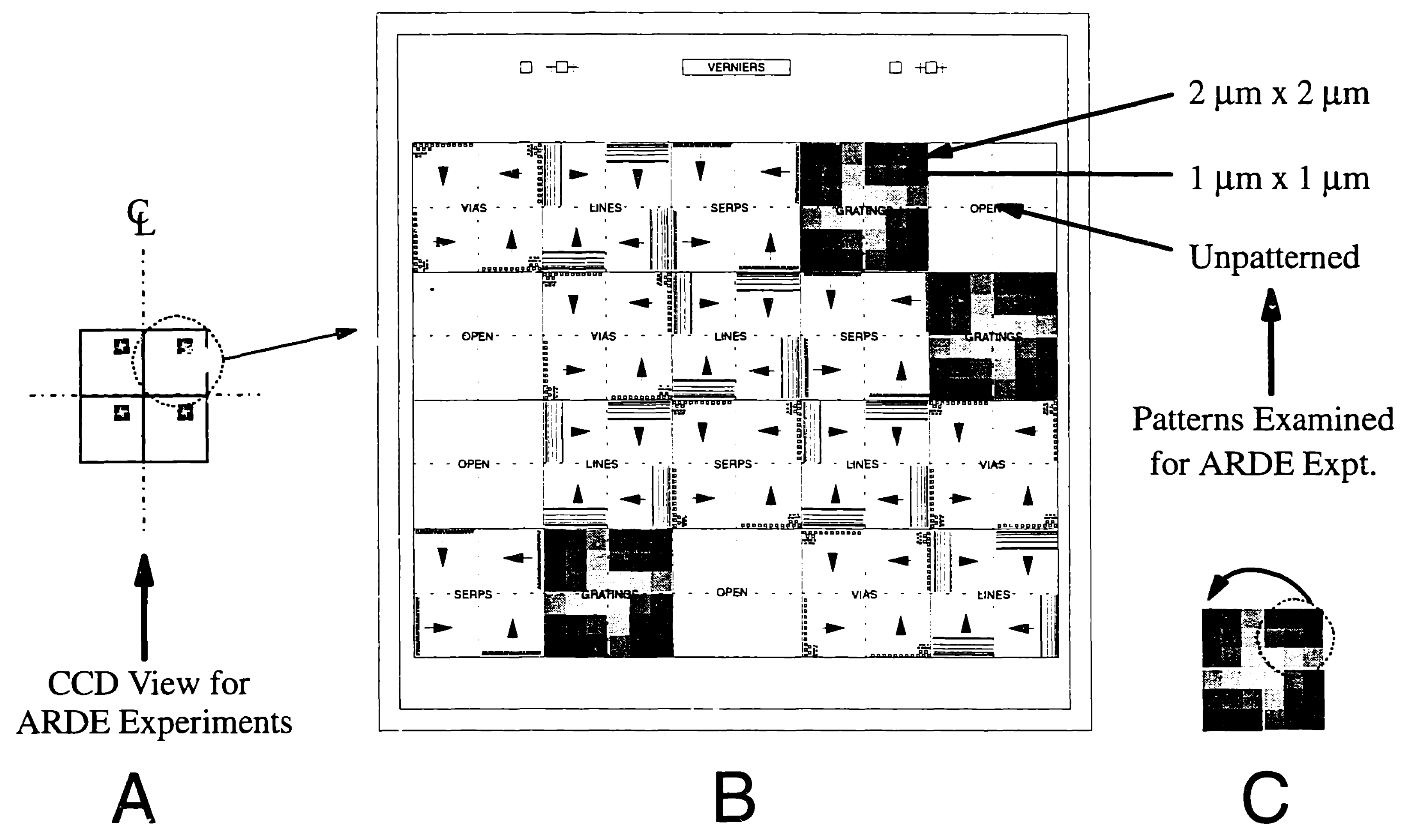


Figure 8-3 : Test Mask Pattern

HBr/Cl_2 chemistry. The nominal values of the process parameters are given in Table 8-1.

Table 8-1

Nominal Process Parameter Values

Independent Parameter	Nominal Value	Units
Pressure	100	mTorr
Gas Composition	75/25 Cl_2/HBr	Percent
Total Gas Flowrate	40	SCCM
Magnetic Field Strength	75	Gauss
Rotation Frequency	1	Hz
Cathode Temperature	20	°C
He Cooling Pressure	10	Torr

8.3 Effect of Process Variables

The effect of process parameters on ARDE was examined. The parameters in this study were: residence time (total flow rate), pressure, magnetic field strength, cathode temperature, feed gas composition, and chamber history/seasoning. The effect of each of these variables is discussed in the following sections.

8.3.1 Residence Time (Total Flow Rate)

The average residence time for gas in the etcher is defined as the ratio of the

chamber volume to the volumetric flowrate of gas. The total flowrate of gas into the etching chamber was varied between 13 and 116 SCCM; all other parameters were as listed in Table 8-1.

The etching rate was measured for features with aspect ratios of 0.0, 0.75, and 1.5. The ratio of the etching rate for the different features to the etching rate for unpatterned polysilicon (open field) (here after, etch rate ratio or ERR) is plotted as a function of aspect ratio for total gas flowrates of 13, 20, 40, and 116 SCCM in Figure 8-4. Two experiments for 40 SCCM are shown to give an indication of measurement repeatability.

A strong dependance of ARDE on residence time is seen in Figure 8-4. At low residence times (high flow rates), the etching is essentially independent of etching rates; At long residence times (low flowrates), the etching rate decreases with increasing aspect ratio showing what is traditionally called RIE lag. As indicated in Figures 8-4, measurements of etching rate variations of a few percent can be observed with this technique; such small variations could not be observed by the analysis of SEM photographs. Flowrate also has an affect on the absolute etching rate (Figure 8-5). The etching rate decreases from around 6300 Å/Min at 116 SCCM to around 4300 Å/Min at 13 SCCM.

8.3.2 Pressure

Changing process pressure has two main effects on the etching process. First, decreasing the pressure decreases the residence time because of the larger apparent volumetric gas flowrate due to a larger gas expansion at lower pressures. Second,

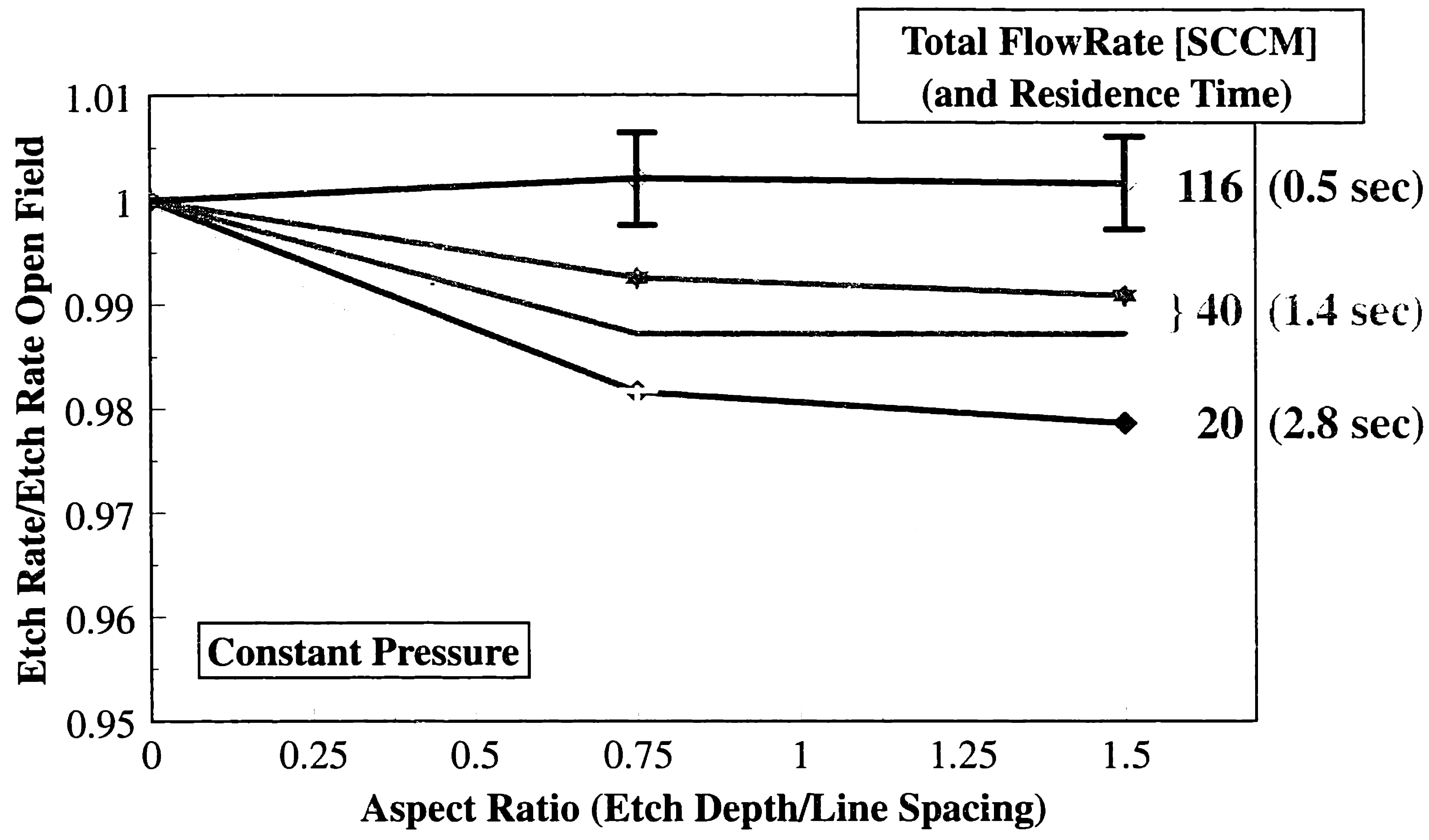


Figure 8-4: Feature Etching Rate Relative to Unpatterned Polysilcion (Open Field) Etching Rate as a Function of Feature Aspect Ratio for Several Different Flowrates

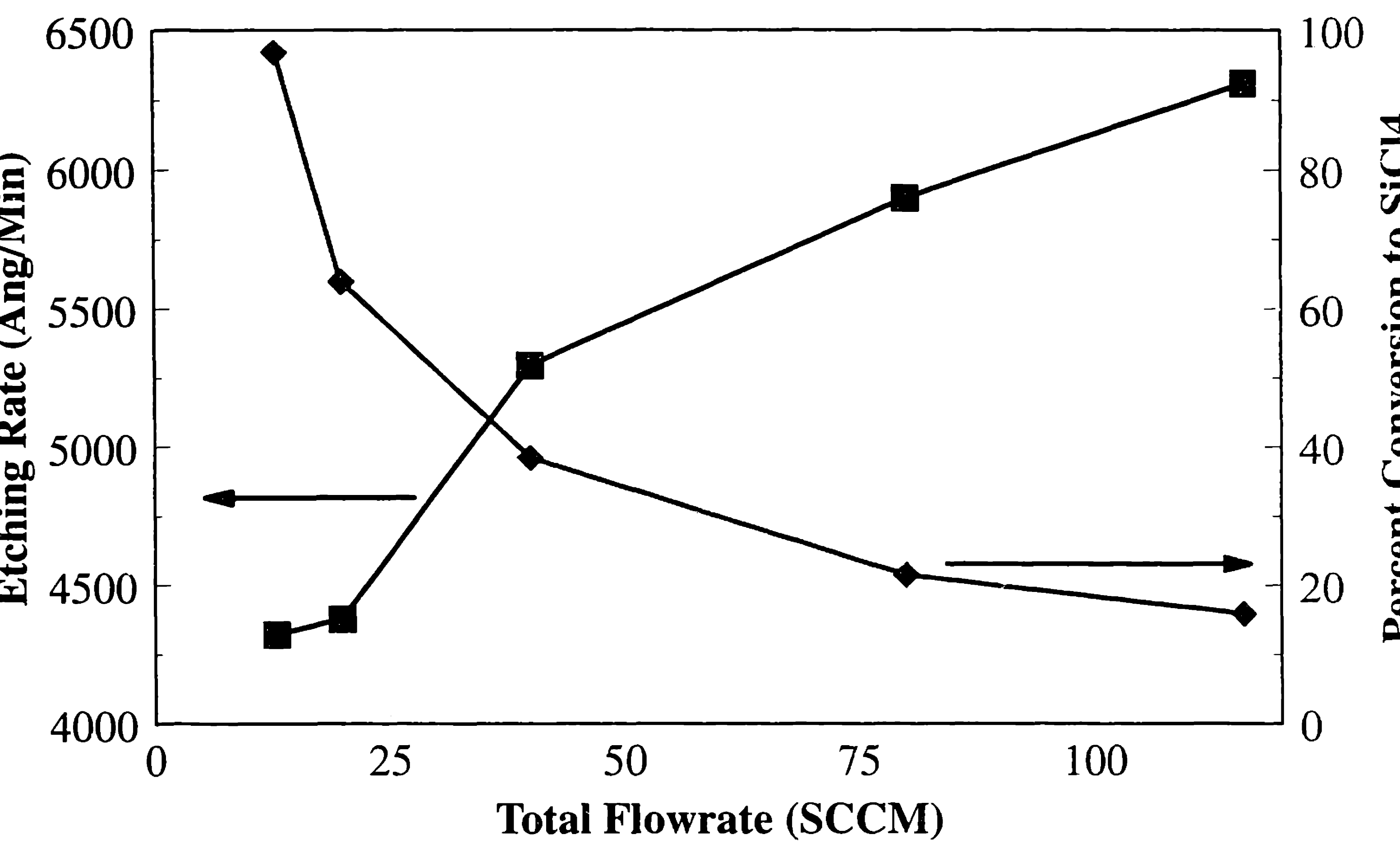


Figure 8-5: Effect of Total Gas Flow Rate on Average Etching Rate.
Also shown is the Required Conversion to Support the Measured Etching Rate.

decreasing the pressure makes the plasma sheath collisionally thinner, leading to a more directional IAD. Both of these effects confound the effect of pressure on ARDE.

Pressure was varied between 10 and 250 mTorr and the magnetic field strength was 0 Gauss; all other parameters are as listed in Table 8-1. The results are plotted in Figure 8-6 which shows the ERR as a function of AR for pressures of 10, 100, 175 and 250 mTorr. Two runs are shown at 100 mTorr taken approximately 1 month apart. At lower pressures, the etching is almost independent of aspect ratio, while at higher pressures, it is strongly dependent on aspect ratio. The scale on this figure should be compared to that of Figure 8-4. Flow rate variation showed a maximum ARDE effect on the order of 4 percent, while pressure showed a maximum of around 10 percent. Even with this relatively large effect, measuring this effect via SEM photographs would still be difficult.

8.3.3 Magnetic Field Strength

Changing process pressure changes both the residence time and the sheath thickness as mentioned in Section 8.3.2. The effect of residence time was discussed in Section 8.3.1 as caused by changing the total flow rate. Changing the magnetic field strength can be used to change the sheath thickness; the sheath thickness decreases as the magnetic field strength increases [Kuypers and Hopman, 1990].

The effects of magnetic field strength on ARDE are shown for magnetic fields of 0 and 75 Gauss, at pressures of 100 and 250 mTorr, in Figure 8-7. All other parameters are as listed in Table 8-1. Also shown on this plot are the error bars for the two

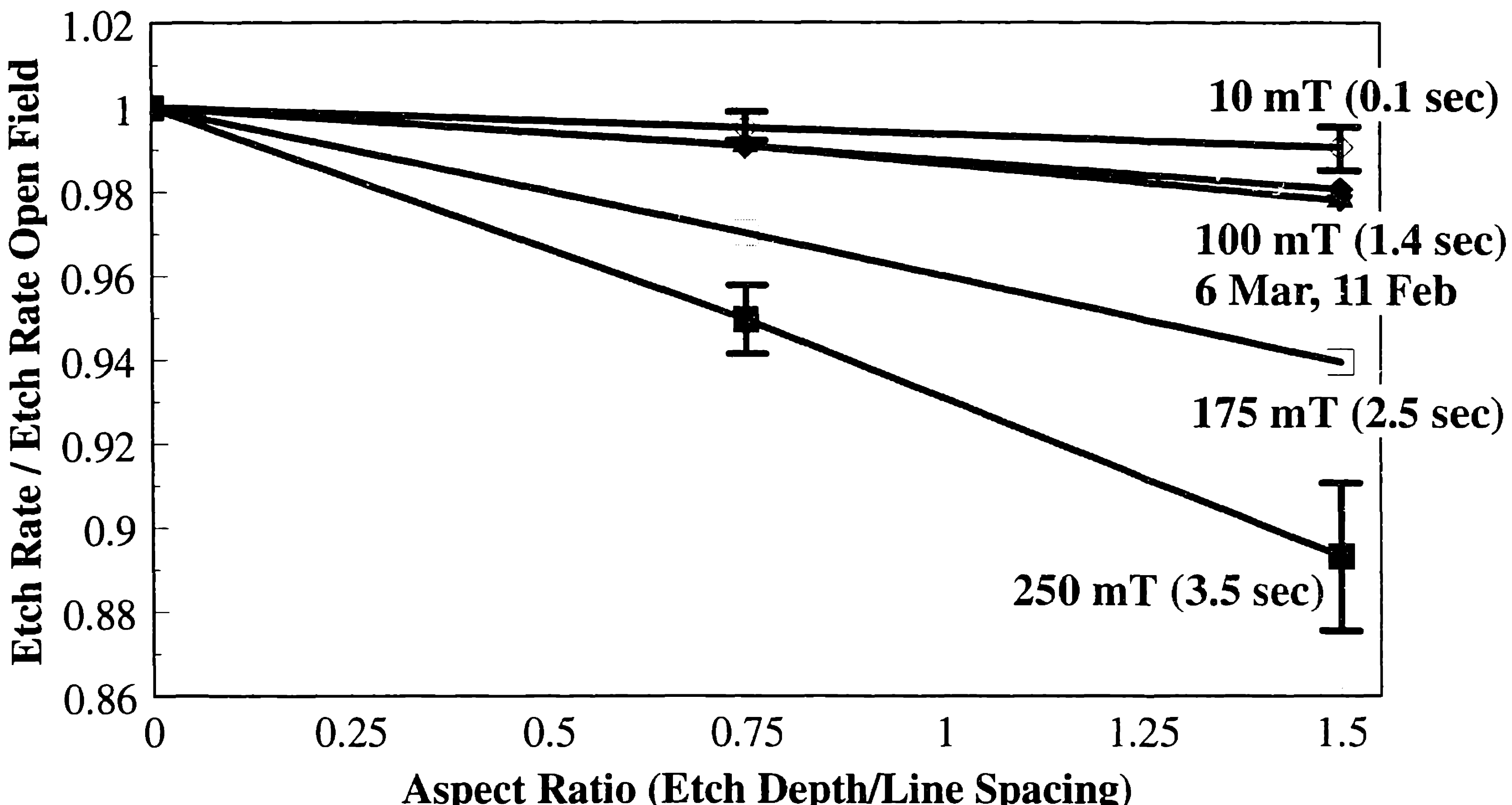


Figure 8-6: Effect of Chamber Pressure on Aspect Ratio Dependent Etching.

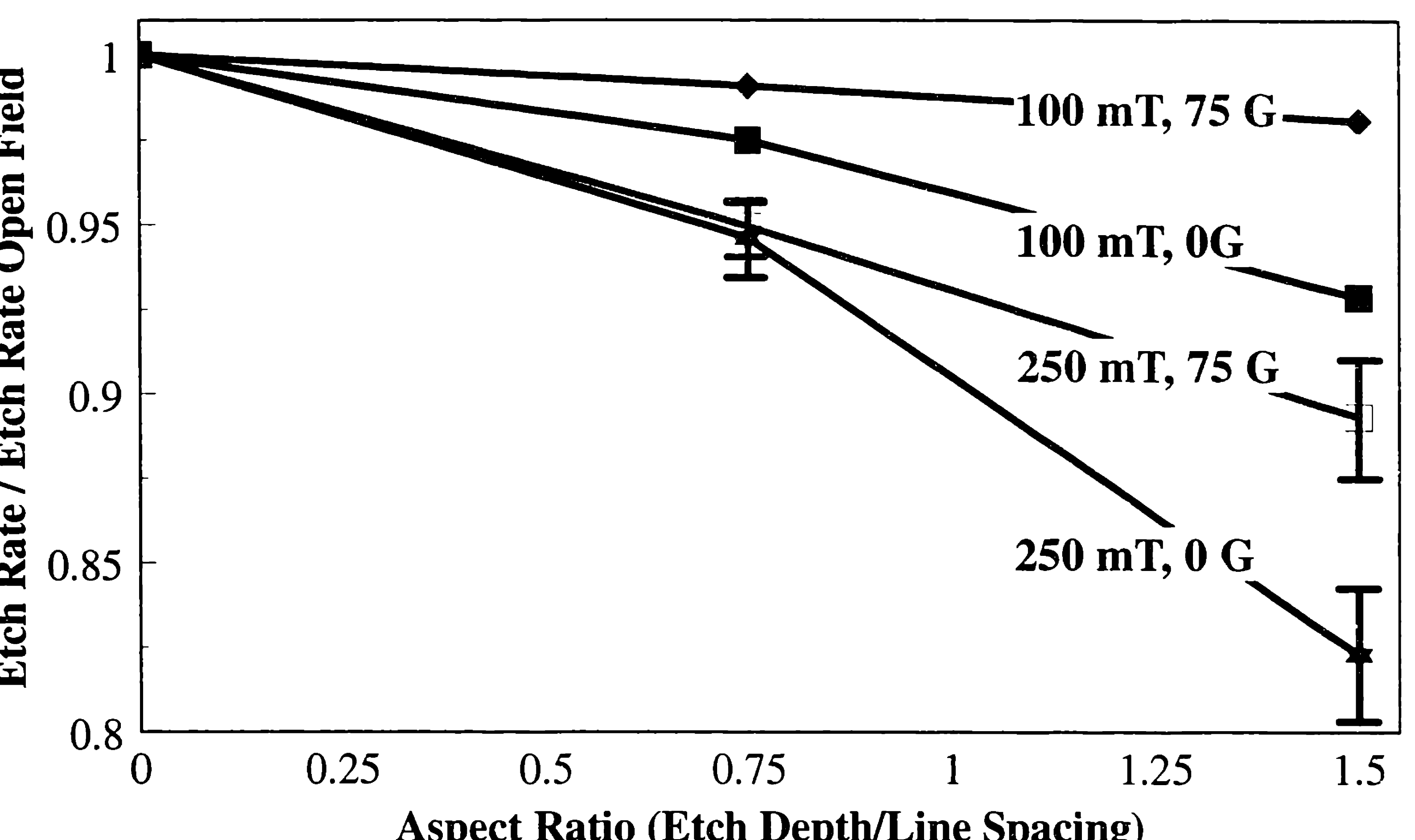


Figure 8-7: Effect of Magnetic Field Strength on Aspect Ratio Dependent Etching.

experiments at 250 mTorr. At both 100 and 250 mTorr, the effect of magnetic field strength is evident. Increasing the field strength decreases the observed ARDE effect. At the extreme condition of 250 mTorr and 0 Gauss, the maximum ARDE effect was on the order of 17 percent.

8.3.4 Temperature

To control wafer temperature, an FTS RC-210C recirculating chiller (FTS Systems, Stone Ridge, NY) was used to cool the cathode pedestal upon which the wafer is clamped during etching; a temperature range of -75°C to +40°C could be obtained. The cathode temperature was varied from -30°C to +35°C; other conditions were as listed in Table 8-1, except that a magnetic field strength of 0 Gauss was used.

The effect of cathode temperature on ARDE is shown in Figure 8-8, which is a plot of ERR as a function of AR for temperatures of +35°C, +20°C, +5°C, -10°C, -25°C, and -30°C. At higher temperatures (+35°C and +20°C), the etching process exhibits RIE Lag in which the smaller features etch more slowly. However, as the temperature is decreased, the etching process exhibits "Inverse RIE Lag" where small features etch more rapidly. This trend continues as the temperature is reduced further, with the largest degree of inverse RIE lag being exhibited at the lowest temperatures. This can be seen in Figure 8-9, which is a plot of the slope of the ARDE curves (Figure 8-8) as a function of temperature. The larger the slope, the larger the RIE lag or inverse RIE lag effects. Negative slopes indicate RIE lag, while positive slopes indicate inverse RIE lag. The transition point (zero slope) is estimated to be around 3°C.

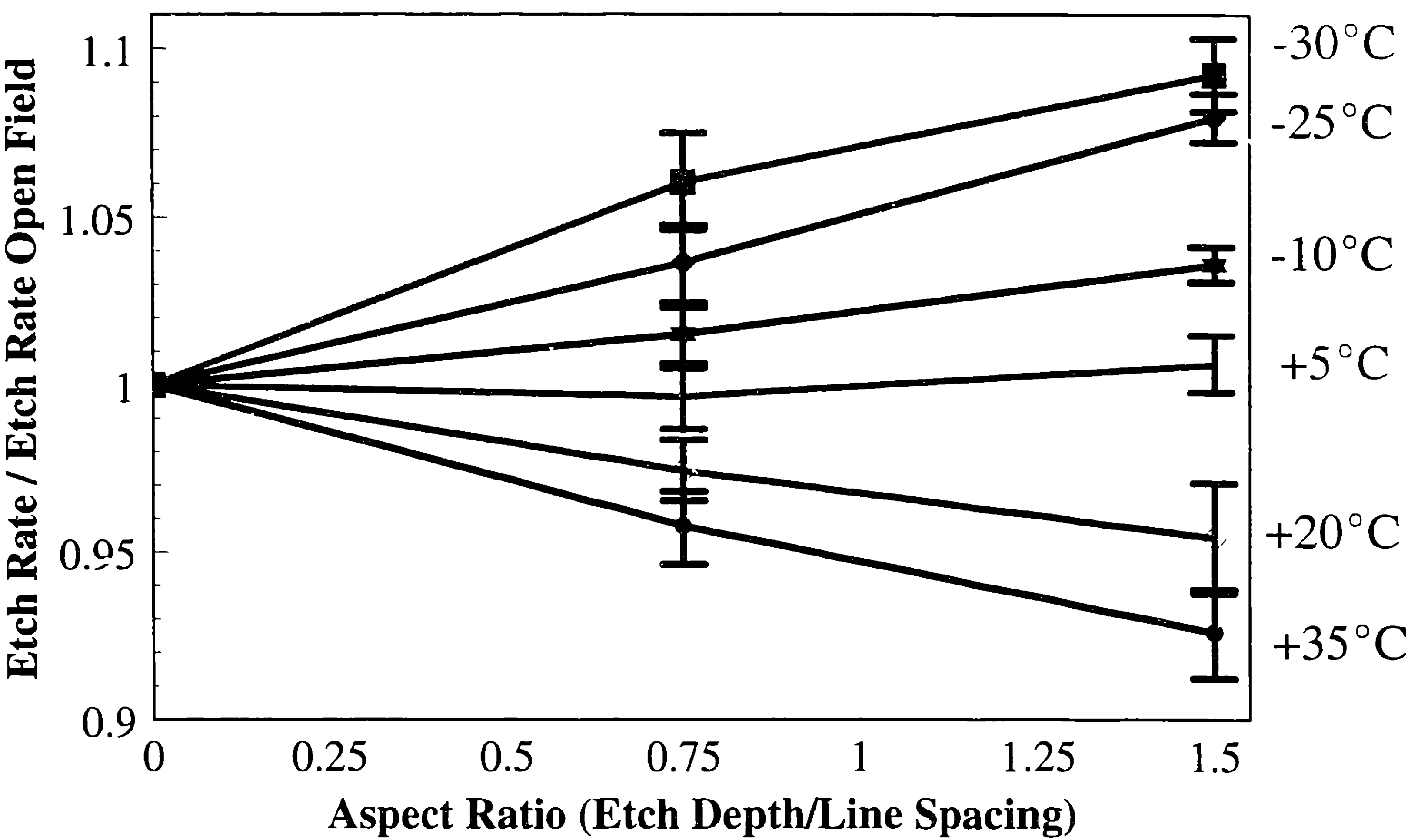


Figure 8-8: Effect of Cathode Temperature on Aspect Ratio Dependent Etching

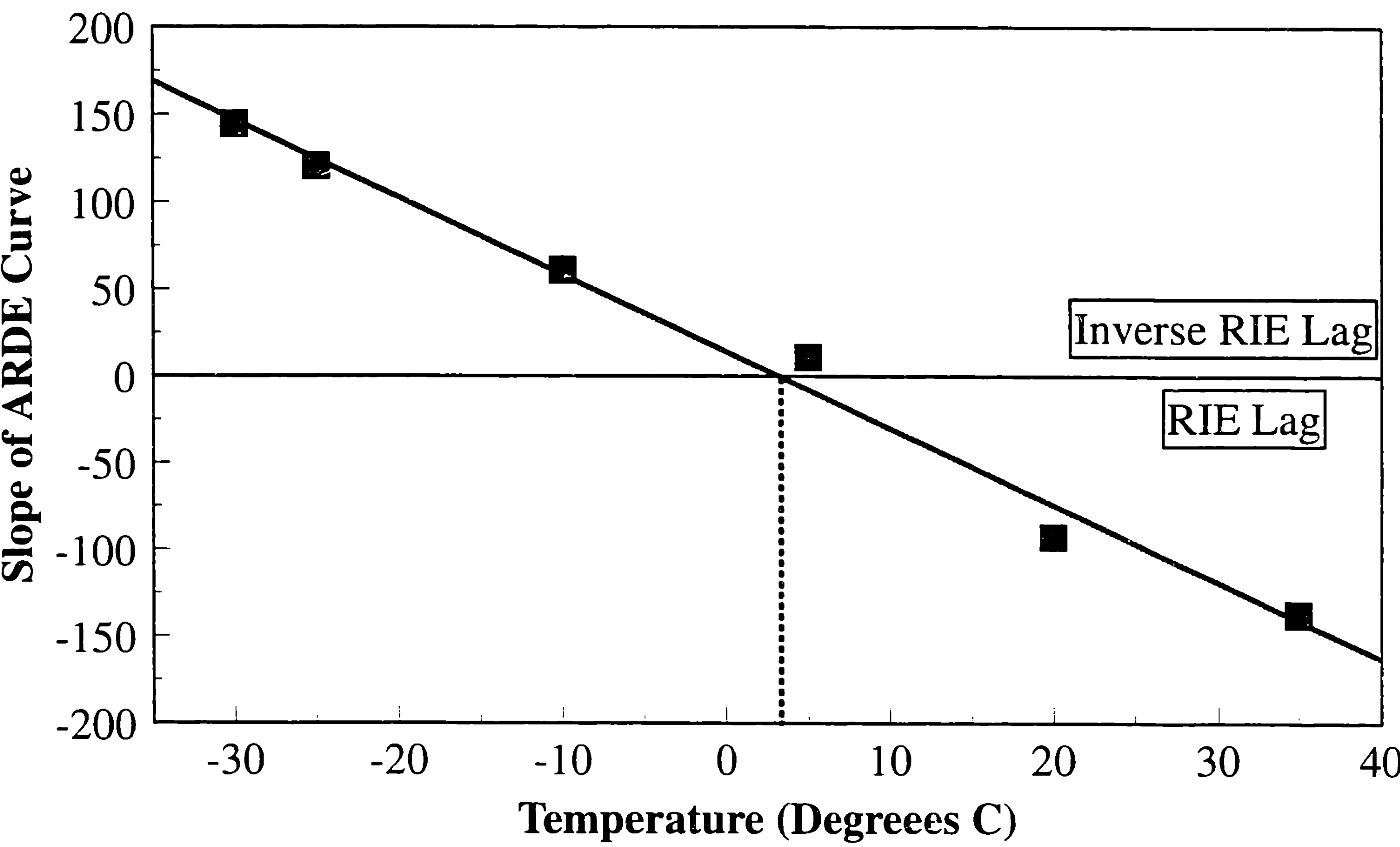


Figure 8-9 : Slope of ARDE Curve (Figure 8-8) showing Transition from "RIE Lag" (Negative Slope) to "Inverse RIE Lag" (Positive Slope).

The absolute etching rates (in Å/Min) is plotted in Figure 8-10 instead of the ERR as a function of AR for cathode temperatures of +20°C, +5°C, -10°C, and -25°C. Reducing the electrode temperature lowers the etching rate of all features, but reduces the etching rate more rapidly for the lower aspect ratios. The apparent activation energies calculated from this data is listed in Table 8-2 as a function of feature aspect ratio.

Table 8-2

Aspect Ratio Dependence of Activation Energy

Aspect Ratio	Ea (Kcal/Mol)	Ea (eV)
0	-0.76	-0.033
0.75	-0.56	-0.024
1.5	-0.34	-0.015

8.3.5 Feed Gas Composition

The composition of the feed gas was varied between 100% HBr and 100% Cl_2, maintaining the other etching parameters as listed in Table 8-1. The ERR is plotted as a function of AR for compositions (%Cl2 / %HBr) or 100/0, 75/25, 50/50, 25/75, and 0/100. At the extremes of pure Cl_2 and pure HBr, inverse RIE lag is observed. At both 25% and 50% Cl_2, little ARDE is observed; at 75% Cl_2, RIE lag is observed. These observations are summarized in Figure 8-12, which is a plot of the slope of the ARDE curves from Figure 8-11 as a function of the gas phase composition. The transition from

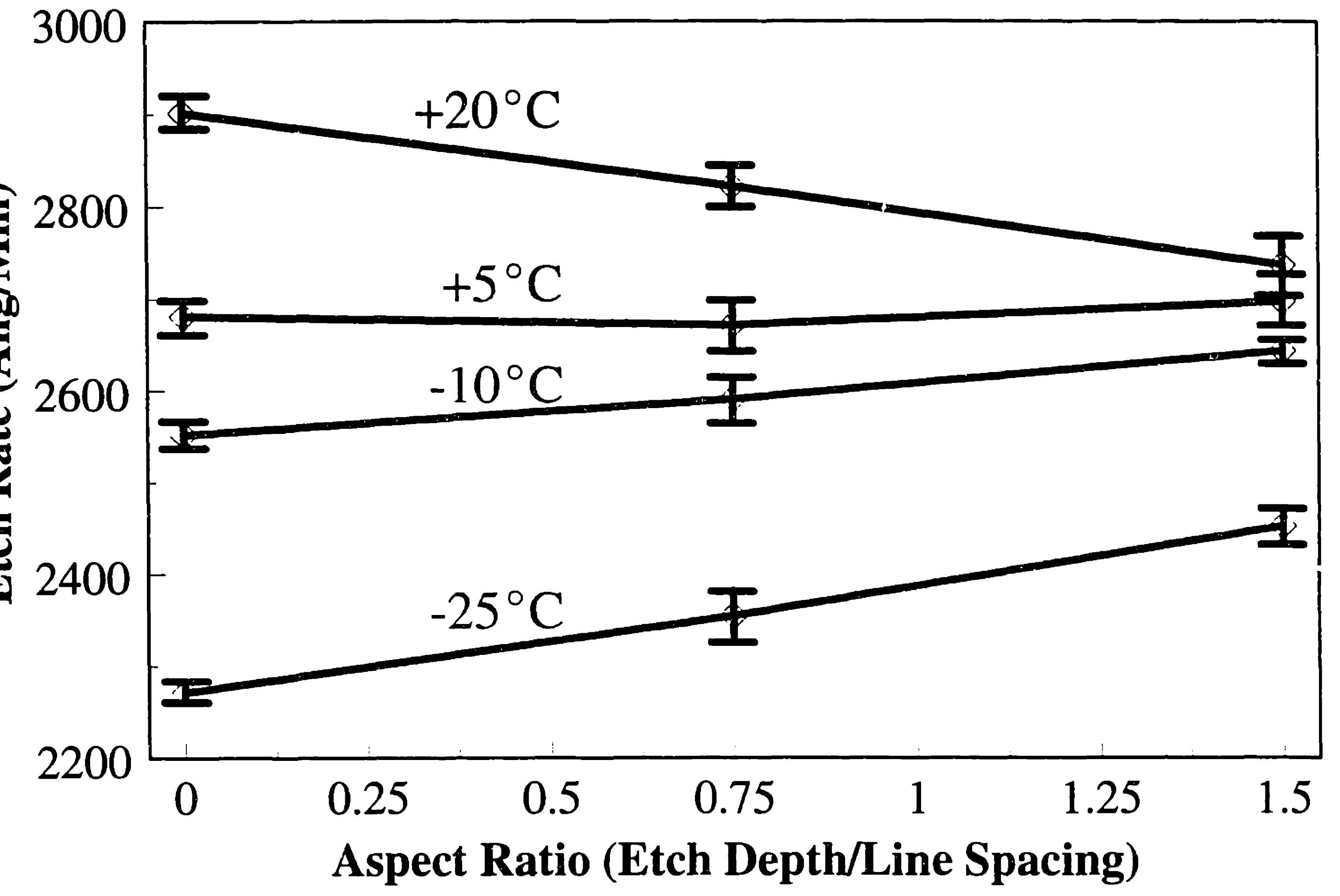


Figure 8-10: Etching Rate as a Function of Aspect Ration for Several Cathode Temperatures.

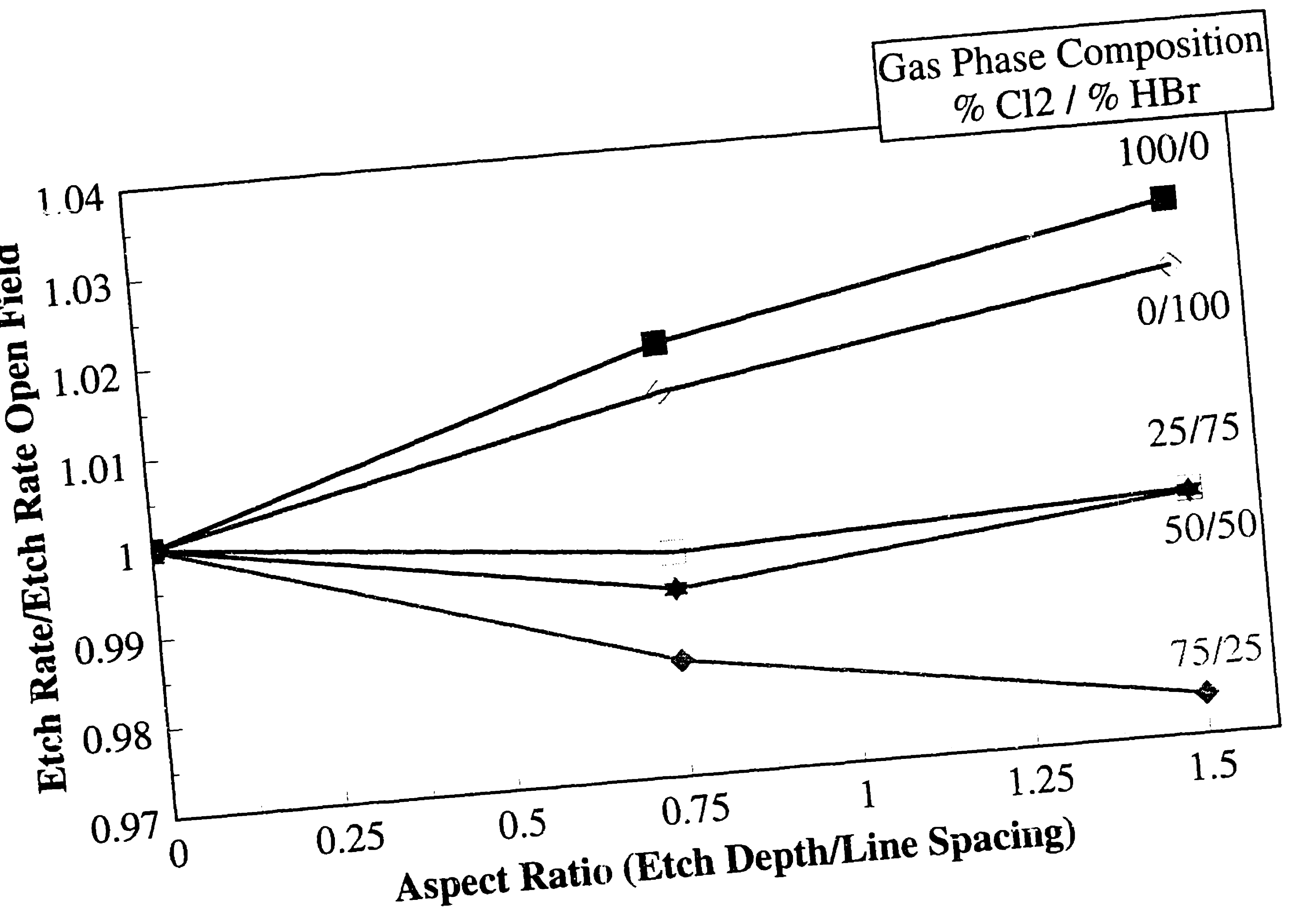


Figure 8-11: Effect of Gas Phase Composition on ARDE.

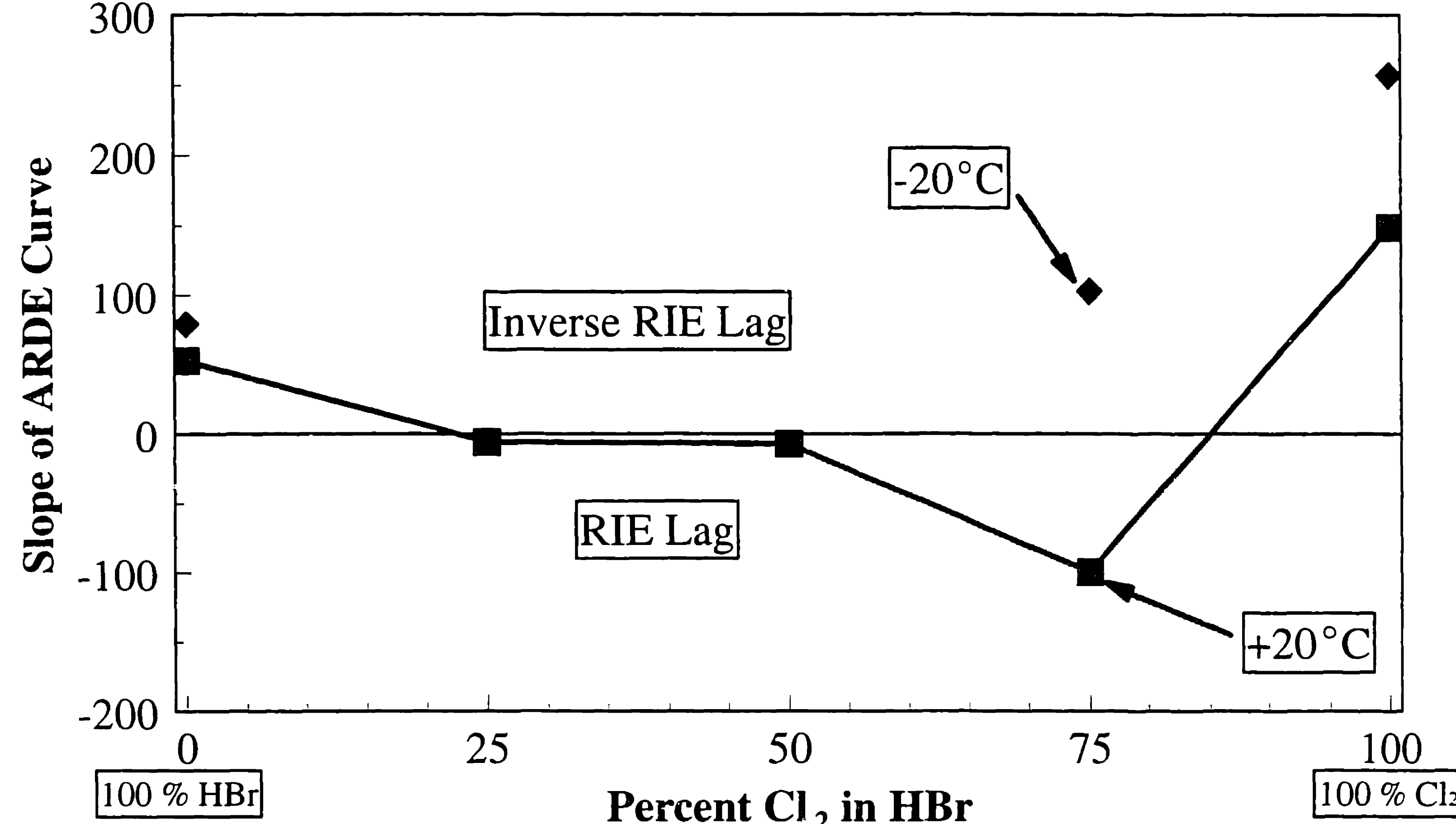


Figure 8-12: Slope of ARDE Curve (Figure 8-11) showing the Effect of Gas Phase Composition on ARDE. Both "RIE Lag" (Negative Slope) and "Inverse RIE Lag" (Positive Slope) Are Observed as Compositon Varies. Etching Temperature was +20°C (Squares). The Diamonds are for etching at -20°C. Point at 75% Cl2 from Figure 8-9.

inverse RIE lag to RIE lag and back as the composition is varied from pure HBr to pure Cl_2 is evident in this plot. The line with the solid square markers is for etching at +20°C; the etching response at -20°C is shown with solid diamond markers. Experiments were conducted for pure HBr and pure Cl_2 under these conditions; the etching response at 75% Cl_2 was estimated from the curve of Figure 8-9. In all cases, decreasing the temperature increases the amount of inverse RIE lag. A comparison of the ERR as a function of AR at the two temperatures is shown in Figure 8-13 for both pure HBr and pure Cl_2.

8.3.6 Chamber History/Seasoning

The condition or state of the etching chamber surfaces has an affect on the etching process. Sawin *et al.* (1985) studied the etching of heavily phosphorus-doped polysilicon in a Cl_2 discharge, where they observed that the etching rate depended on the electrode material. Both an anodized aluminum electrode and a CF_3Cl discharge-seasoned stainless steel (SS) electrode showed the same behavior for the etching rate as a function of frequency. However, an acid-cleaned stainless steel electrode showed a significant difference at frequencies above 1 MHz; the etching rate increased with frequency for the anodized Al and seasoned SS electrodes while the etching rate decreased with frequency for the acid-cleaned SS electrode. The difference in the etching rate and in the etching profiles was attributed to an increased recombination or neutralization of the etchant species on the acid-cleaned SS. Deposition of a polymer film on chamber walls was found to be crucial for reproducible etching rates [Hayes *et al.*, 1989] for the CH_4/H_2 etching of InP; Furthermore, non-zero etching and deposition rates were obtained at 0%

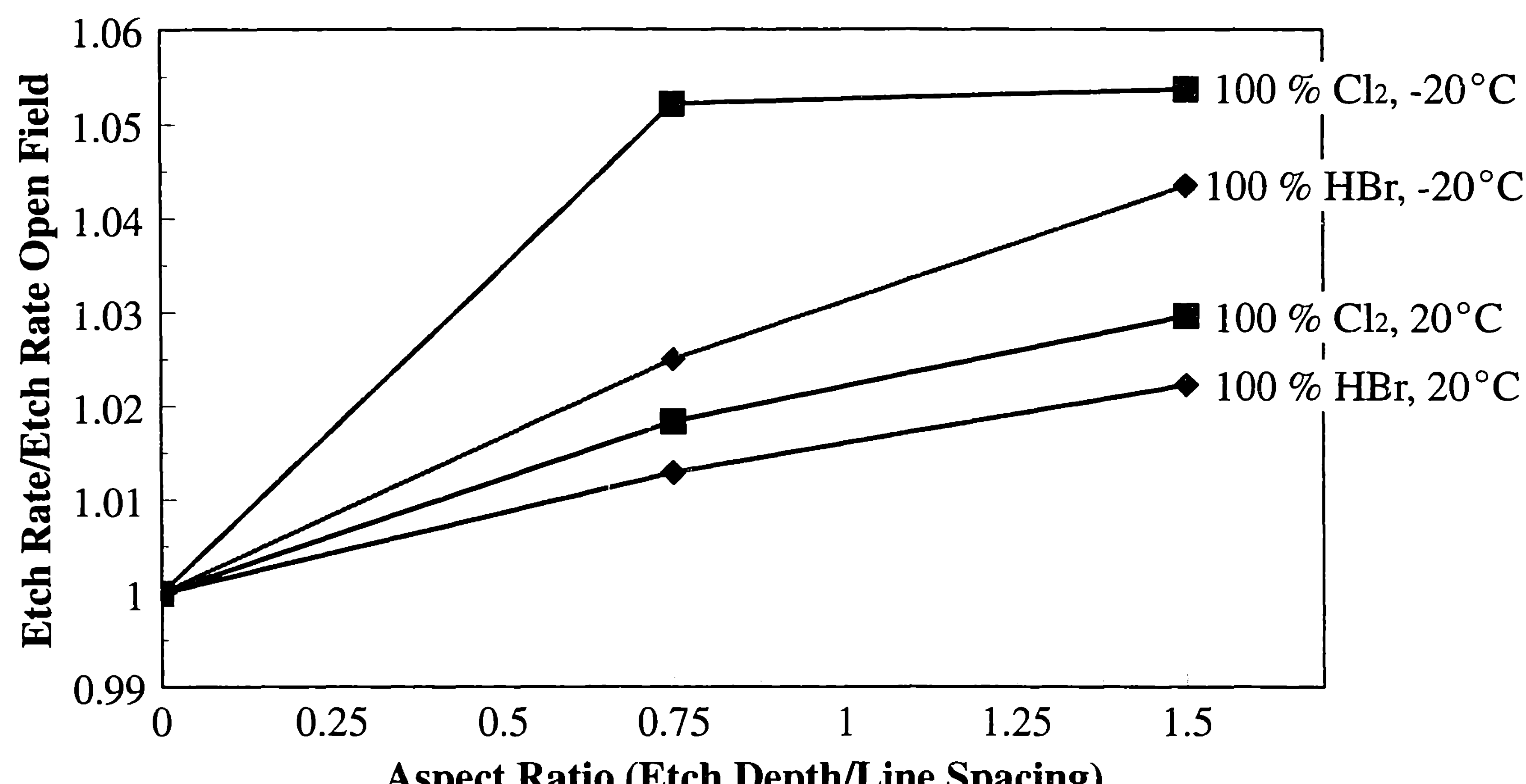


Figure 8-13 : Effect of Temperature And Composition on ARDE.

CH_4 due to the reactions of the plasma with the polymer deposited on the chamber walls. These observations point out the importance of the chamber surface condition. Despite its significance to the etching process, little is known about these effects.

The effects of chamber condition are shown in Figure 8-14. The line with the dark square markers represents the ARDE measured initially. Shortly there after, an NF_3/HBr trench etch process was run in the chamber. The chamber was cleaned using an Ar discharge and an O_2 discharge for approximately 1 hour each. The lines shown with the diamond and star symbols represent measurements of ARDE made after the chamber clean. Large scatter was observed in this data, than in other data; this is consistent with the observations of Hayes *et al.* [1989]. After approximately a month of heavy use, the ARDE was again measured in this chamber, and is shown as the line with hollow square markers. This line corresponds very well with the one from before the chamber clean, showing that the chamber walls have returned to the same state as they previously were.

8.4 Plasma Characterization

To better understand some of the ARDE phenomena observed in Section 8.3, a study was undertaken to characterize the plasma. Plasma OES was measured for several species to provide insight into plasma chemistry. A simple CSTR model demonstrated that the dominant plasma chemistry could be accounted for.

8.4.1 Plasma Optical Emission Spectroscopy

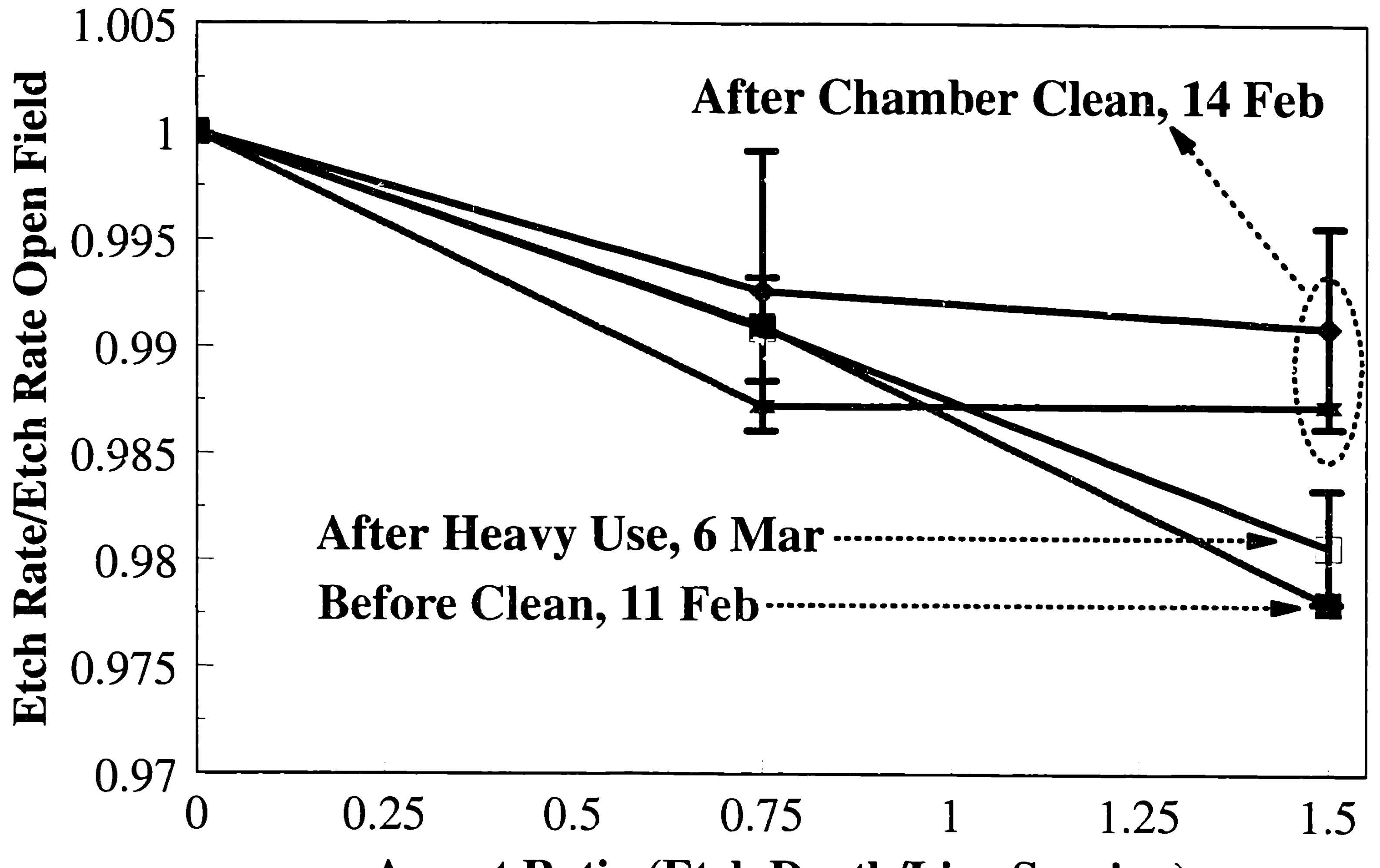


Figure 8-14: Affect of Chamber History/Seasoning on ARDE

Changing the gas residence time in the reactor (Figure 8-4) changes the gas phase chemistry. In order to understand what changes are occurring, optical emission spectroscopy (OES) was used to monitor several plasma species. Plasma optical emission was sampled using an optical fiber bundle butted up against a side optical view port, and run into a Jobin Yvon HR 640 0.64 m monochromator with two 1800 groove/mm diffraction gratings; one had a blaze of 2500 Å, and the other had a useable range of 4500-10000 Å. A Hamamatsu R955 photomultiplier tube was used as the detector. The monochromator was controlled and the photomultiplier was powered using an Instruments SA Spectra Link. The photomultiplier current was measured with a Keithley 427 current amplifier. Optionally, a Jarrell-Ash 0.25 m monochromator with a 1200 groove/mm grating and with a Tracor-Northern TN-6200 1024 element linear diode array could be used.

A number of species were identified in the plasma; they included atomic, molecular and ionic species and are listed in Table 8-3, along with their wavelengths, identification references, and actual wavelength used on the HR 640 monochromator (1)[Reader and Corliss, 1980], (2)[Pearse and Gaydon, 1984], (3)[Tiller and Sameith, 1990], (4)[Ibbotson *et al.*, 1983]. Argon was added to the discharge for use as an actinometer [Coburn and Chen, 1980] at both 5 and 10 percent of the total flowrate. There was no difference in the trends in the actinometric curves for both Ar concentrations so 5% was used.

Table 8-3

Plasma Species and Optical Emission Lines

Species	Wavelength (Å)	Reference	HR 640 Setting (Å)
Cl	7547.072	(1)	7547.2
Cl^+	3827.59	(1)	3829
Cl_2	5950.5	(2)	5950.5
Si	2881.579	(1)	2882
SiCl	2807.2	(2),(3)	2807.9
$SiCl_2$	3300	(3)	3300
Br	7512.96	(1)	7512.7
Br^+	4704.85	(1)	4705.8
Br_2^+	5766	(4)	5763.5
Ar	7503.869	(1)	7503.7

An example of the identification process used to assign the peaks in the optical emission spectrum is shown in Figure 8-15, which is a plot of the plasma OES spectrum from 2800 Å to 2840 Å. The two peaks visible are from the doublet splitting of the $X^2\Pi_r$ ground state in the $B^2\Delta$ - $X^2\Pi_r$ electronic transition and compares well with published spectra [Tiller and Sameith, 1990]. The maximum in the band structure was used as the SiCl emission line. Other emission lines and bands were identified by examination of the plasma spectrum in the vicinity of known peaks, and by checking for overlapping peaks from other emission sources.

The ratios of the optical emission intensity for the various species to that of Ar as a function of the total gas flowrate for cathode temperatures of +20°C and -25°C are

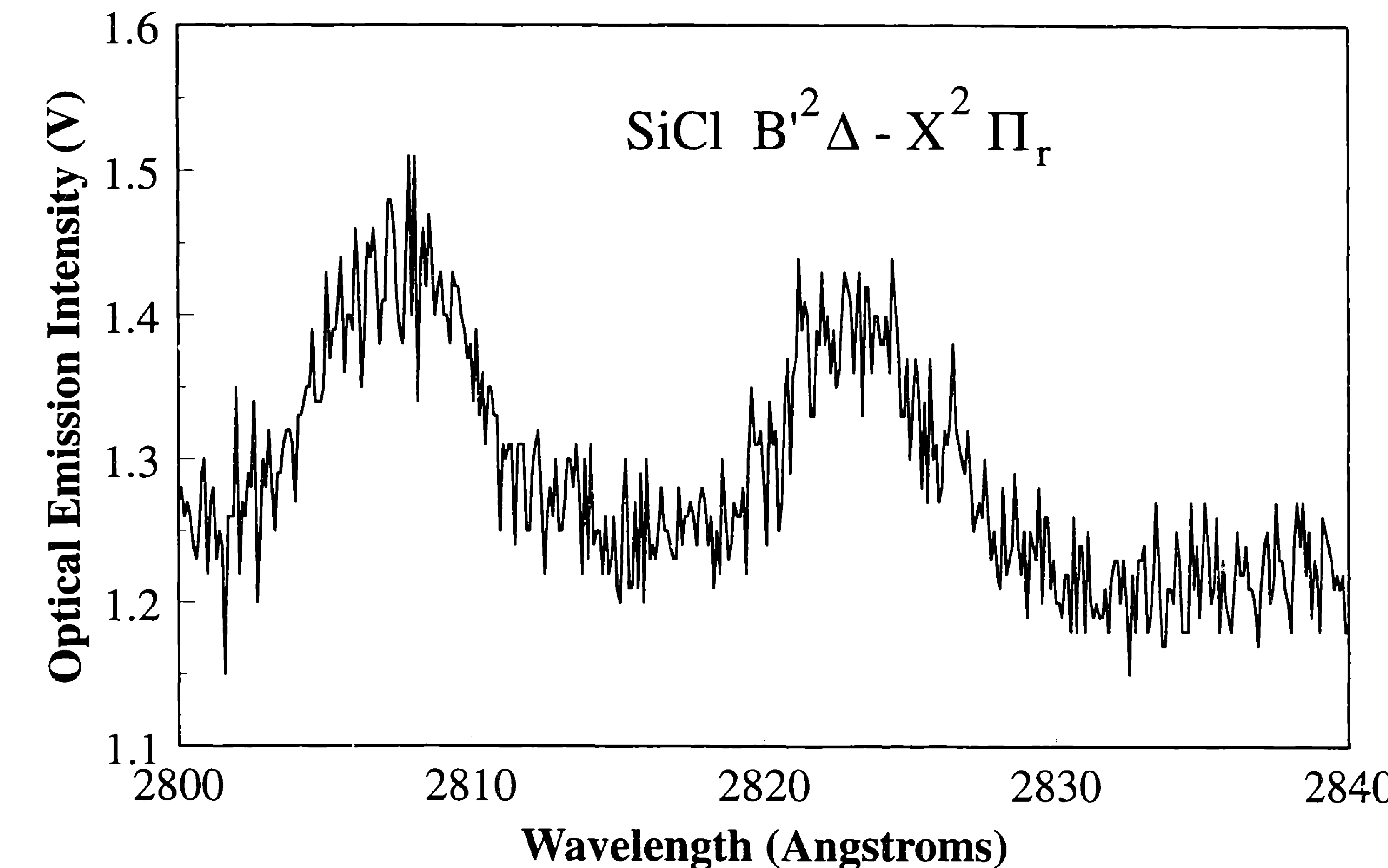


Figure 8-15 : Optical Emission Signal for SiCl Radical. Doublet Splitting of the $X^2\Pi_r$ ground state is visible in this $B'^2\Delta$ - $X^2\Pi_r$ Electronic Transition.

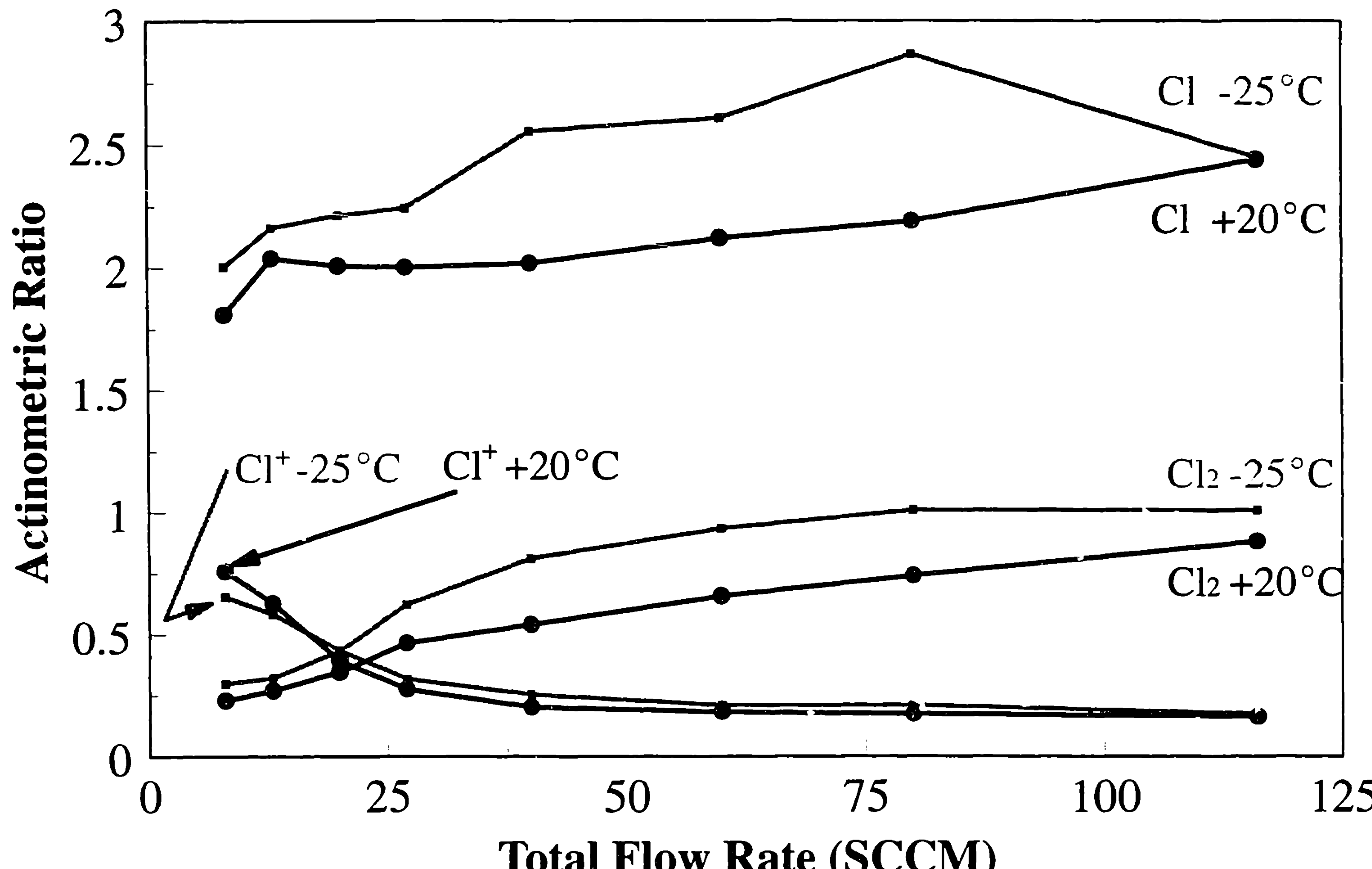


Figure 8-16A: Actinometric Ratios as a Function of Total Flowrate for Cathode Temperatures of +20°C and -25°C. Cl, Cl_2, and Cl^+OE Ratioed to Ar OE.

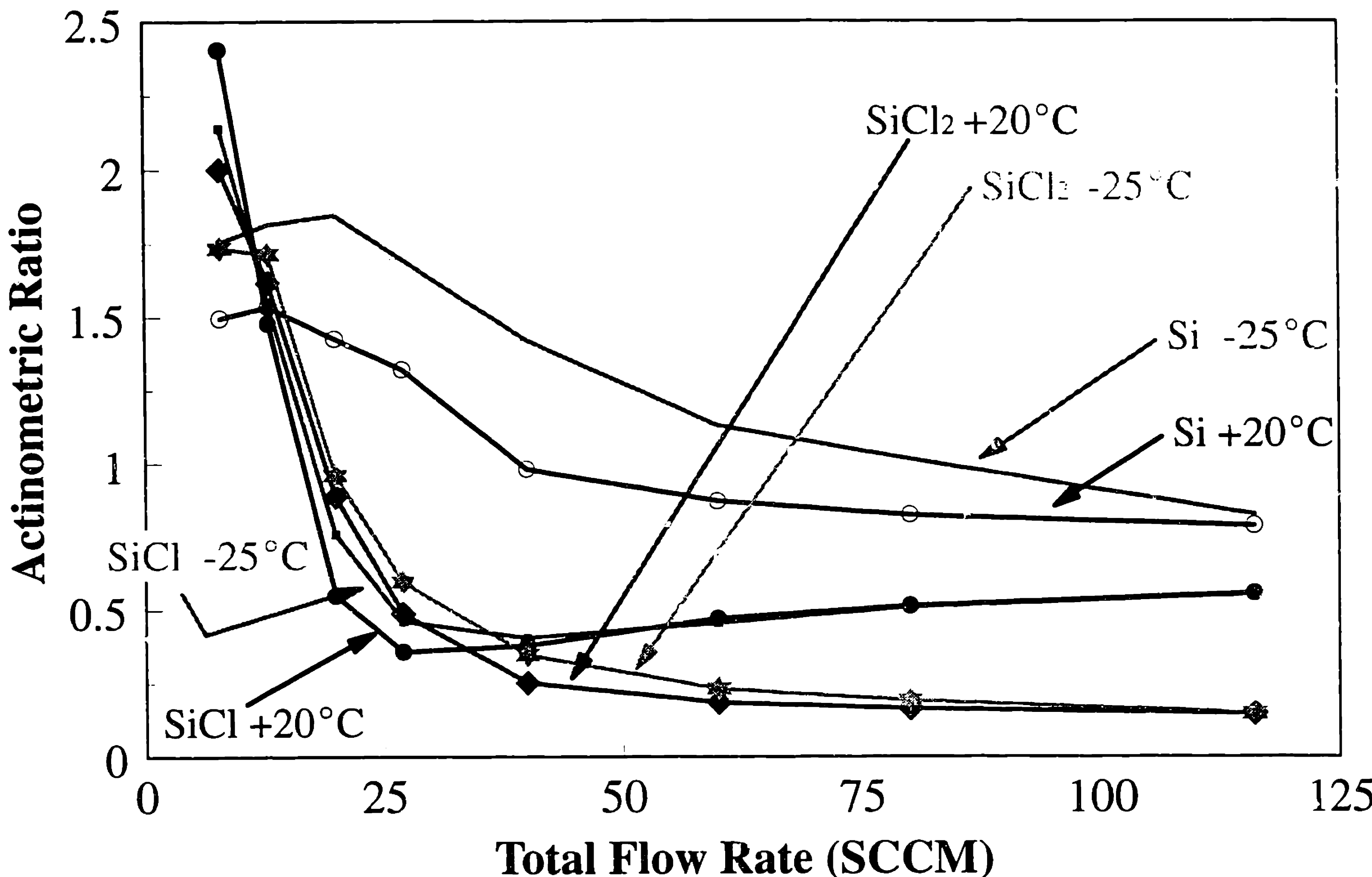


Figure 8-16B: Actinometric Ratios as a Function of Total Flowrate for Cathode Temperatures of +20°C and -25°C. Si, SiCl, and $SiCl_2$ OE Ratioed to Ar OE.

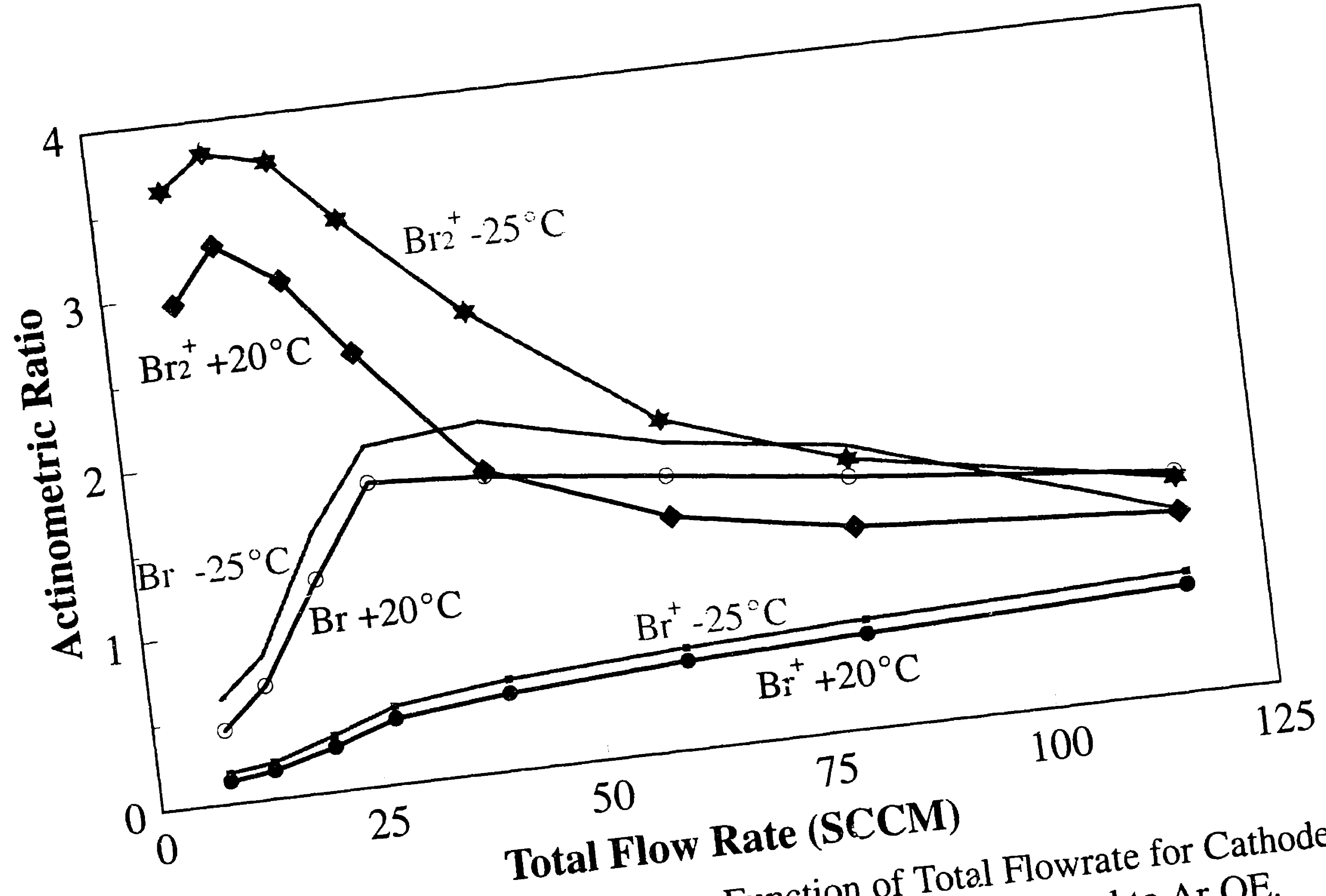


Figure 8-16C: Actinometric Ratios as a Function of Total Flowrate for Cathode Temperatures of +20°C and -25°C. Br, Br^+, Br_2^+ OE Ratioed to Ar OE.

plotted in Figures 8-16A, 8-16B, and 8-16C. These data were taken at the conditions of Table 8-1, with a magnetic field strength of 0 Gauss; the feed gas composition was maintained at 75% Cl_2 and 25% HBr over the entire range of total gas flowrates. There is little difference for the spectra at the different temperatures. Cl and Cl_2 both show a decrease with decreasing flowrate while Cl^+ shows an increase (Figure 8-16A). The decrease in Cl and Cl_2 can be explained by decreasing amounts of Cl_2 introduced into the reactor as the flowrate drops, and by the increased fraction consumed in etching reactions. Si, SiCl, and $SiCl_2$ all show a large increase as the flowrate is decreased (Figure 8-16B), despite the fact that the etching rate decreased by about 30% at the same time (Figure 8-5), decreasing the amount of Si and Si-containing species being released into the plasma. Br^+ decreases with decreasing flowrate, while Br_2^+ increases; Br initially increases and then decreases sharply with decreasing flowrate (Figure 8-16C).

8.4.2 Plasma Chemistry

The optical emission data of the previous section yield some insight into the chemistry occuring within the plasma. When etching blanket polysilicon over oxide wafers, OES shows a large change in the Cl actinometric signal at end-point, while the Br signal is essentially unchanged (Figure 8-17); the change in the Cl signal is because of an increase in Cl concentration caused by either: (1) slower consumption of Cl on SiO_2; or (2) slower recombination of Cl on SiO_2. Due to the behavior of Cl at end-point and the large fraction of Cl in the plasma, we can assume that Cl is the dominant etchant. The etch product is expected to be $SiCl_4$, and possibly $SiCl_2$ [Rossen and Sawin, 1987].

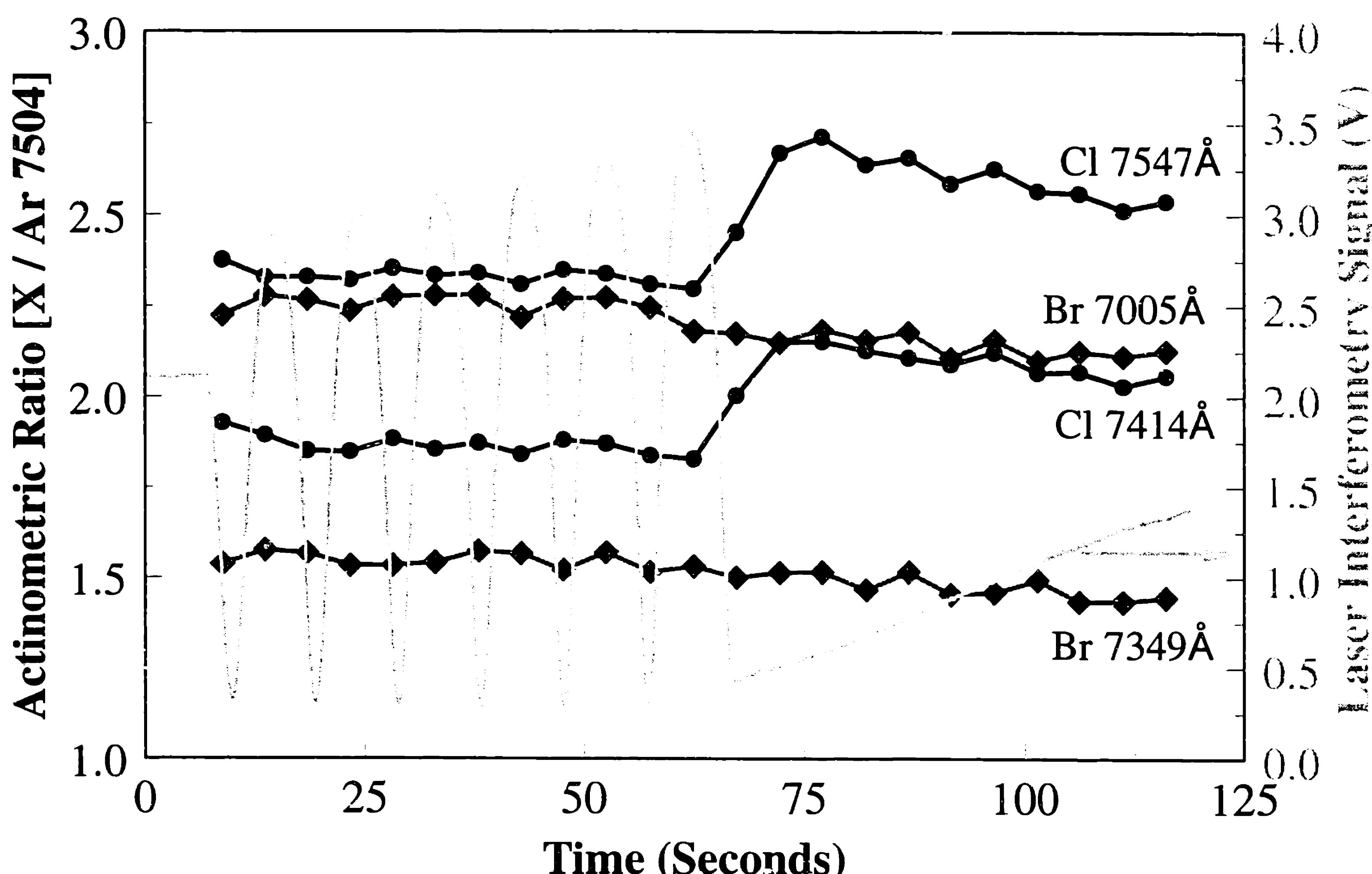


Figure 8-17: Actinometric Ratios for Cl (7414Å and 7547Å) and Br (7005Å and 7349Å) with Simultaneous Laser Interferometry Trace. Cl Rise Corresponds to End-Point.

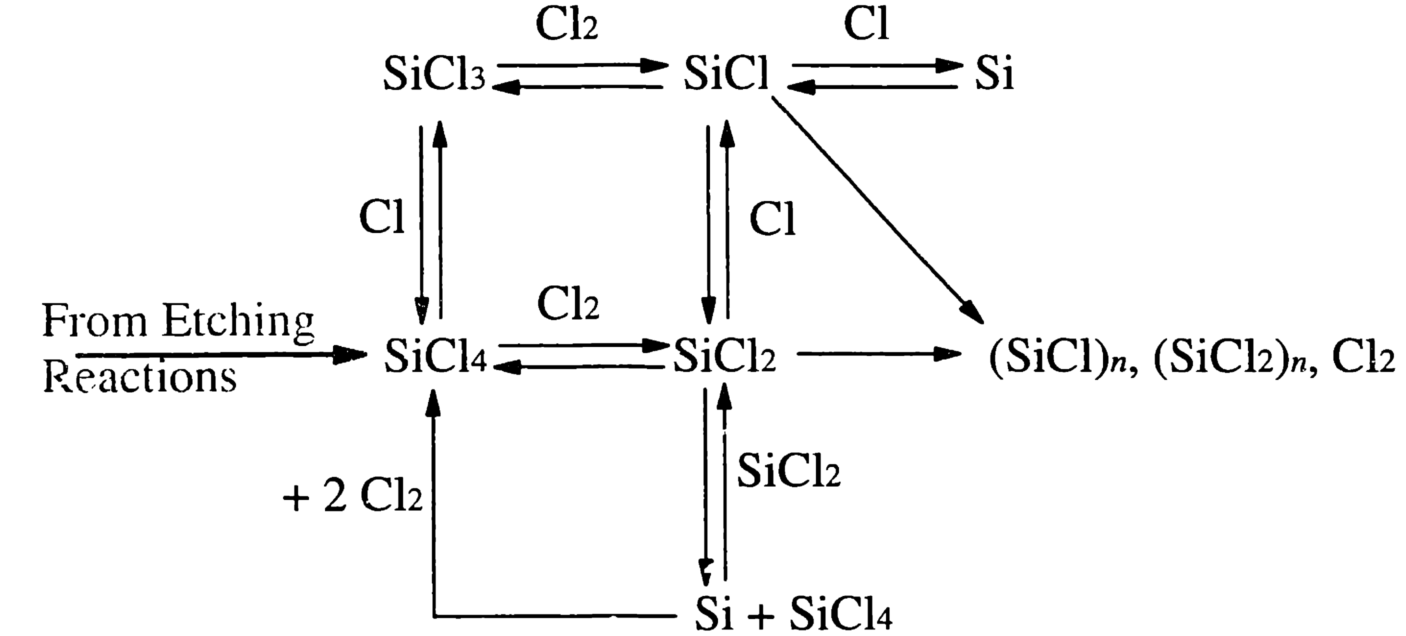


Figure 8-18: Reaction Network For Decomposition of $SiCl_4$ to Form $(SiCl_x)_n$ Polymers. After [Tiller and Sameith, 1990], [Tiller *et al.*, 1984].

$SiCl_4$ and other etch products undergo reaction in the gas phase. Reaction networks for the decomposition of $SiCl_4$ are summarized in Figure 8-18 [Tiller and Sameith, 1990],[Tiller *et al.*, 1984]. The important point in this reaction network is that $SiCl_4$ decomposes to yield SiCl and $SiCl_2$ radicals. These radicals can readily deposit on surfaces in the plasma and can polymerize to form silicon chloride polymers, $(SiCl)_n$ and $(SiCl_2)_n$. Such polymers have been observed on chamber walls after etching [Tiller and Sameith, 1990],[Sameith *et al.*, 1986]. After conducting experiments at very long residence times (low flowrates), we observed a polymer deposition on the walls of the etching chamber also. $SiCl_4$ can also react with oxygen present to form SiO_2, Si_2OCl_6, and Si_3OCl_8, which can also deposit on the wafer [McNevin, 1990],[McNevin, 1991].

The dissociation of $SiCl_4$ to yield Si radicals and $SiCl_x$ radicals can occur by mechanisms other than electron impact dissociation. Hydrogen can abstract a Cl to form HCl [Sato and Arita, 1987]. H is present from the dissociation of HBr in the plasma. We have observed emission from HCl at 9152 Å and 7463 Å [Pearse and Gaydon, 1984]; the origin of these emissions from reactions between H and Cl or H and Cl_2 can not be ruled out.

8.4.3 CSTR Model for Plasma Behavior

A simple continuously stirred tank reactor (CSTR) model was constructed for the etching process. This was essentially a mass balance on the etching chamber. The electron-impact dissociation rate and the dissociative attachment rates of Cl_2 were estimated from the work of Rogoff *et al.* [1986] assuming an electron density of

10^{10} cm^{-3}; the formation rate coefficient for Cl was estimated to be on the order of 3.5×10^{-9} cm^3 s^{-1}. Because of the large number of unknown reaction rates for the reaction network of Figure 8-18, only the electron impact dissociation of $SiCl_4$ to $SiCl_2$ and Cl_2 was included in the gas phase chemistry. Si was assumed to leave the surface entirely as $SiCl_4$, with the rate of $SiCl_4$ production coming from experimentally measured etching rate data. The sticking coefficient for Cl_2 was estimated to be 0.3 [Barker *et al.*, 1983], and that for Cl was assumed to be 1.0.

A CSTR model assumes that the reactor is "well mixed." That is, there are no spatial concentration gradients. Use of the CSTR model is justified when several criteria are met. The first criterion is that the Peclet number, *Pe*, be small. The Peclet number is defined as

$$Pe \equiv \frac{v}{(D/l)} \qquad \textbf{(8-1)}$$

where v is the convective gas velocity, D is the gas phase diffusivity, and l is the characteristic length scale for diffusion. *Pe* represents the ratio of the convective transport velocity to the diffusive transport velocity. As $Pe \rightarrow 0$, the reactor becomes more uniform due to rapid diffusion or slow convection. As $Pe \rightarrow \infty$, convection is much faster than diffusion so that concentration gradients exist along the flow path. The limit of $Pe = \infty$ is the limit of a plug flow reactor (PFR). The gas velocity can be estimated from the gas flowrate, chamber pressure and an assumed gas phase temperature of 500 K. The area for gas flow is the annular region at the edge of the cathode that leads to the turbo pump port. The diffusivity is estimated to be 1130 cm^2/s at 100 mT, 298 K [Chang *et al.*, 1985]. The diffusion length is 5 cm for an entire wafer (center to edge) or .5 cm for

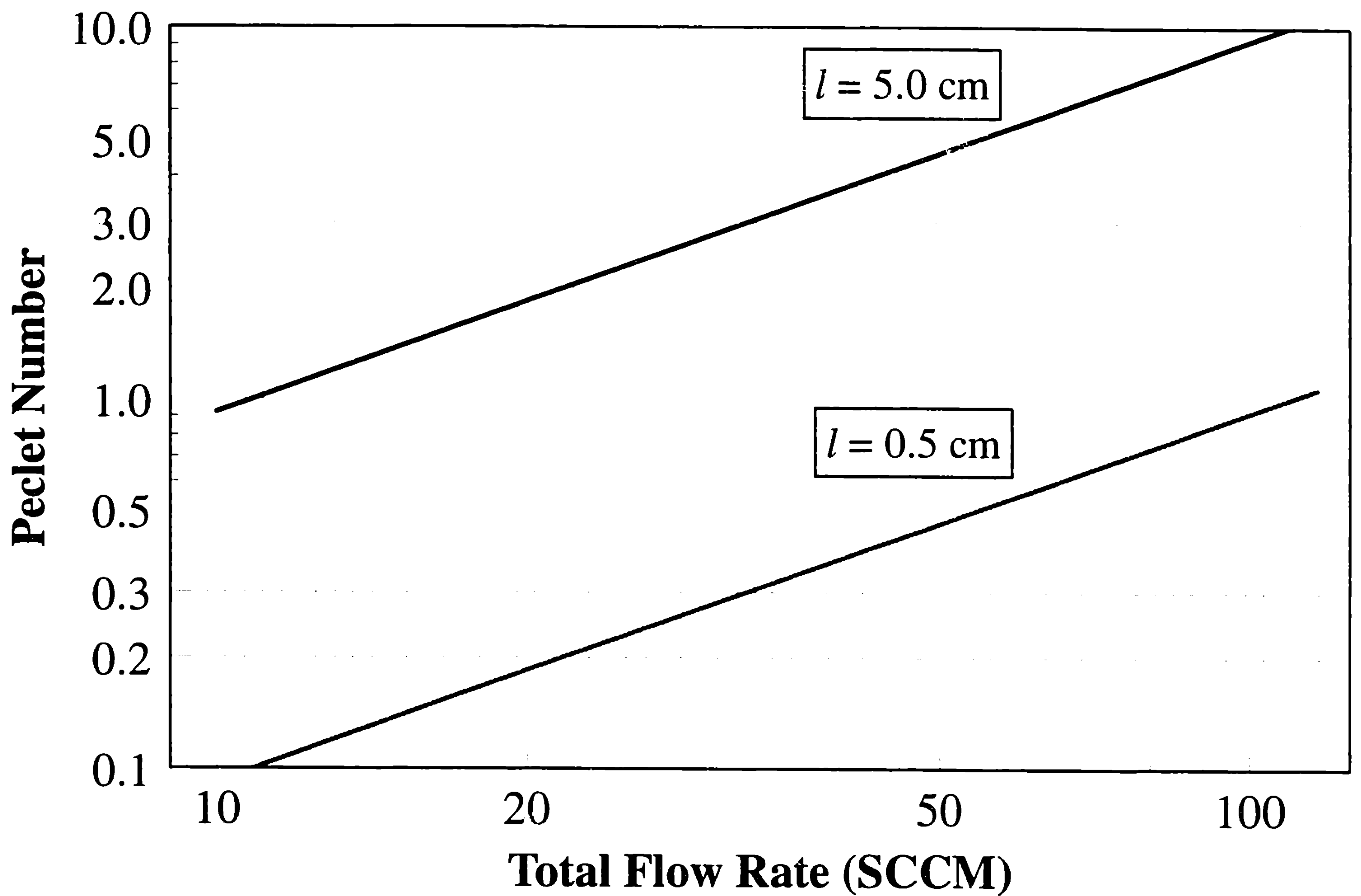


Figure 8-19: Calculatd Values of the Peclet Number as a Function of Flowrate for an Entire Wafer ($l = 5.0$ cm) and for a Die ($l = 0.5$ cm).

a die. The calculated maximum values of *Pe* are shown in Figure 8-19. The calculated values indicate that there may be concentration gradients across the wafer. The value of v used represents a worst case number. The actual flow velocity across the wafer is likely to be lower than our estimate. Still, the gradients may exist.

Because reactions are occurring on the wafer surface and not just in the gas phase, a second criterion must be met. The Damköhler number, *Da*, must be small; it is defined as

$$Da \equiv \frac{k_{surf}}{(D / l)} \tag{8-2}$$

where k_{surf} is the surface reaction rate coefficient. Da represents the ratio of the reaction velocity to the diffusive transport velocity. As $Da \rightarrow 0$, transport is much faster than reaction so that concentration gradients do not occur. For $Da \rightarrow \infty$, reaction is much faster than transport; mass transport controls the reaction rate and significant concentration gradients can occur. Assuming a Boltzman flux to the wafer surface, k_{surf} can be evaluated as

$$k_{surf} = \frac{\gamma \, |\bar{v}|}{4} \tag{8-3}$$

where γ is the surface reaction probability, and $|\bar{v}|$ is the mean speed (Sawin and Reif, 1989). *Da* is evaluated as a function of γ for l = 5.0 cm, 0.5 cm, and 1.0 μm (feature scale) in Figure 8-20. The dotted vertical line on this figure represents an estimate of the reaction probability from the CSTR model (discussed below). This is a lower limit estimate of γ. Thus, across a wafer, *Da* may be approximately 0.1 or greater, thus possibly satisfying the CSTR criterion.

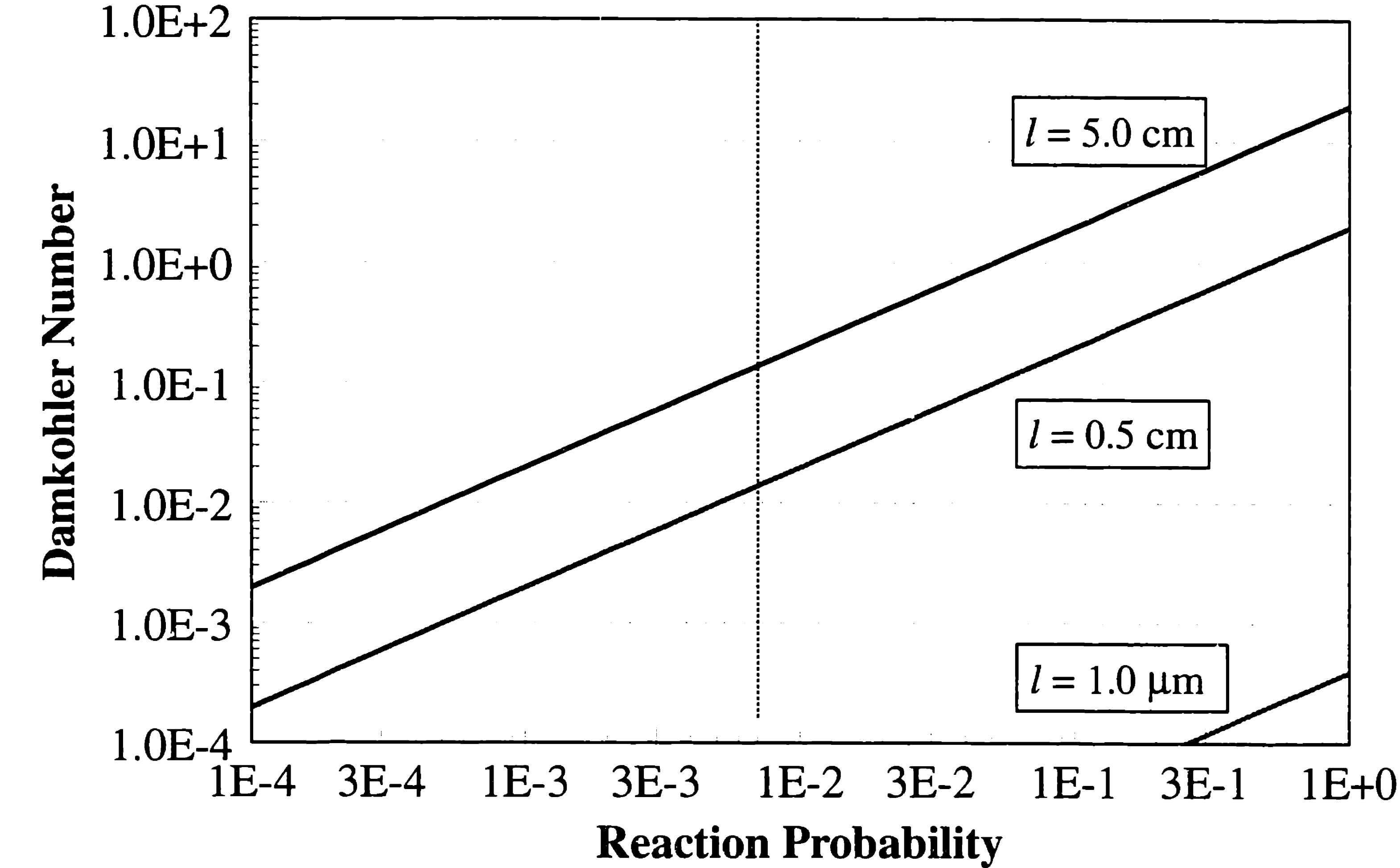


Figure 8-20: Calculatd Values of the Damkohler Number as a Function of Reaction Probability for an Entire Wafer (l = 5.0 cm), for a Die (l = 0.5 cm), and for a Feature (l = 1.0 μm). Also Shown is a Lower Estimate of the Reaction Probability from CSTR Model.

The kinetics for $SiCl_4$ dissociation have not been quantified. However, some of the kinetics of CCl_4 dissociation have been studied. The rate constant for the electron attachment dissociation of CCl_4 to CCl_3 and Cl^- is 5 x 10^{-7} cm^3 $molecule^{-1}$ s^{-1} [Tiller *et al.*, 1983]. The deposition of materials from polymeric decomposition products of $SiCl_4$ is expected to be much lower than that from carbon-based chemistries such as CCl_4 [Tiller and Sameith, 1990]; thus the dissociation rate is expected to be lower. We chose to use a $SiCl_4$ dissociation rate of 10^{-10} cm^3 s^{-1}, which is almost an order of magnitude smaller than the Cl_2 dissociation rate.

Calculated concentrations for Cl, Cl_2, $SiCl_2$, and $SiCl_4$ (for a $SiCl_4$ dissociation rate of 10^{-10} cm^3 s^{-1}, which is significantly slower than the CCl_4 electron-impact dissociation rate of 4.5 x 10^{-8} cm^3 s^{-1} and the CCl_4 electron attachment dissociation rate of 1 x 10^{-7} cm^3 s^{-1} [Gogolides and Sawin, 1989]) are plotted in Figure 8-21 as a function of the total gas flow rate. The Cl and Cl_2 curves show similar trends as the measured values in Figure 8-16A; the $SiCl_2$ curve is remarkably similar to the measured curve of Figure 8-16B. The decrease in Cl with decreasing flowrate is due to the increased conversion of Cl to $SiCl_4$ at low flowrates (Figure 8-5). At high flowrates (116 SCCM), less than 20% of the feed gas is required to support the observed etching rate. However, as the flow rate decreases the required conversion increases, despite the reduced flow loss of species, until at low flowrates (13 SCCM), near 100% conversion is required; most likely, Cl liberated during the electron-impact dissociation of $SiCl_4$ participates in additional etching reactions.

The reaction probability predicted with this model is 7 x 10^{-3}. This may be a little low, but we have neglected both recombination reactions and simultaneous etching and

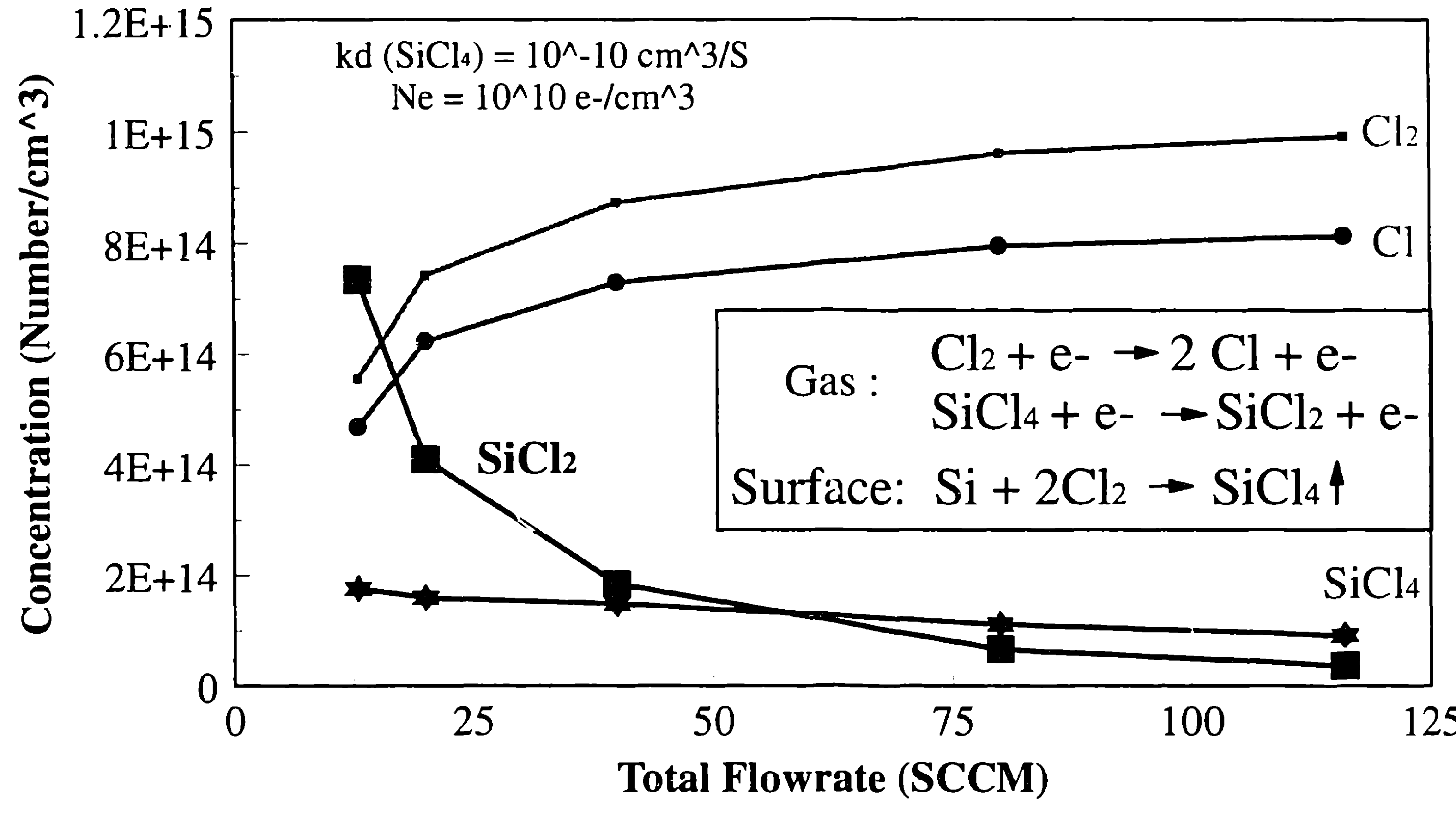


Figure 8-21: CSTR Model Results for Species Concentrations as a Function of Total Flow Rate.

deposition reactions. The model results show that the major trends in the experimental data can be accounted for with this simple model. Changing the $SiCl_4$ dissociation rate changes the absolute concentration of $SiCl_2$ and $SiCl_4$, but not the trends. Similarly, changing the Cl_2 sticking coefficient changes the relative amounts of Cl and Cl_2 but not the trends.

8.5 Discussion

At least two different mechanisms are responsible for the data presented in Section 8.3. The effects of residence time and of temperature can be explained by deposition on the wafer from the plasma. The effects of magnetic field strength and pressure can be explained by changes in the ion angle distribution (IAD) and the resultant change in ion shadowing. The following sections explain the observations made in Section 8.3.

8.5.1 Residence Time and Aspect Ratio Dependent Etching

The dependance of ARDE on etcher residence time (flow rate) can be explained by the transport of reactive etchant species (Cl) and of reactive polymer forming species ($SiCl_{x=1,2}$). At low flow rates, there is a lower concentration of etchant and a higher concentration of polymer forming species. The increased deposition of polymer is seen as a decrease in the line spacing. Line spacings measured from SEM photographs indicated that polymer deposition resulted in smaller spacings as the flow rate decreased

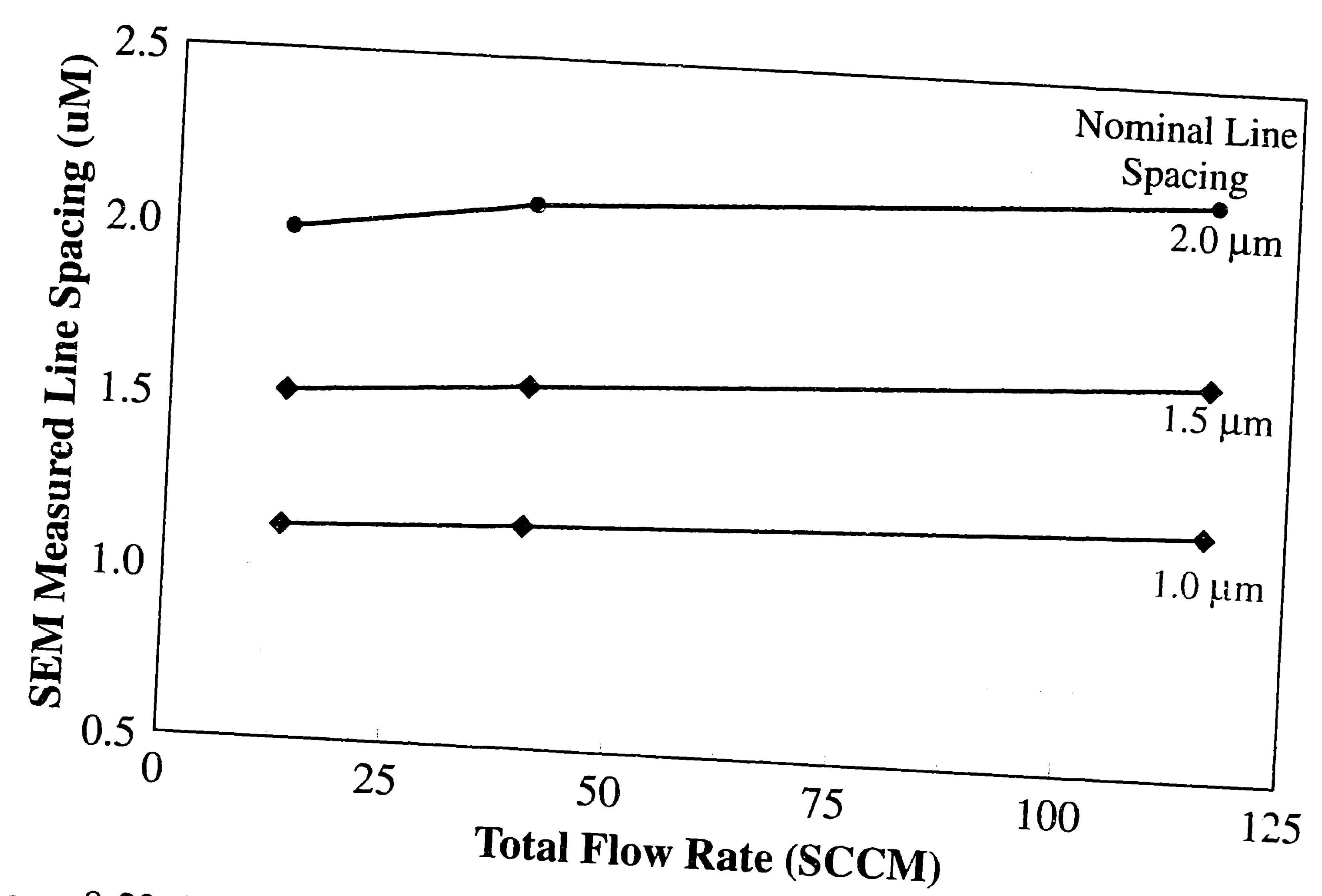


Figure 8-22: SEM Measured Nominal Line Spacings (μm) as a Function of Total Flow Rate (SCCM) for Nominal Line Spacings of 1.0, 1.5, and 2.0 μm.

(Figure 8-22). The reduced Cl levels and the high conversion at low flow rates made reactant transport limitations into features much more important. At high flowrates, $SiCl_4$ was removed from the reactor by flow before it had a chance to appreciably dissociate, leading to lower polymer levels. This, combined with the increased levels of Cl resulted in etching that showed little dependance on AR.

The importance of $SiCl_4$ to polymer deposition is known. Sato and Arita [1987] added $SiCl_4$ to a Cl_2/H_2 discharge to control the shape of the trench. They observed deposited Si films that could easily be removed with HF. The film deposited on our wafers was also easily removed in HF and by O_2 RIE. Sato and Arita [1987] added 3 SCCM of $SiCl_4$ to a 60 SCCM gas feed. The etching rates observed in our experiments (Figure 8-5) indicate a $SiCl_4$ evolution rate ranging from 12.6 SCCM for a 13 SCCM feed gas flow to 18.4 SCCM for a 116 SCCM feed gas flow; these evolution rates are sufficient to provide deposition precursors.

8.5.2 Temperature and Aspect Ratio Dependent Etching

Temperature has a very strong influence on ARDE (Figure 8-8). At low temperatures, inverse RIE lag is obtained; others have seen inverse RIE lag in fluorocarbon systems where there is significant deposition from the gas phase [Vitkavage *et al.*, 1991],[Horwitz *et al.*, 1988]. inverse RIE lag was observed during the high-selectivity (high polymer deposition) CF_4/H_2 etching of SiO_2 by Howritz *et al.* [1988]; it was attributed to the "larger polymer acceptance angle" of low aspect ratio features. This trend was reversed at higher pressures where RIE lag was seen. Similar pressure effects

were observed by Vitkavage *et al.* [1991] who studied the etching of SiO_2 in $CHF_3/Ar/CF_4$ discharges. The deposition of materials from polymeric decomposition products of $SiCl_4$ is expected to be much lower than that from carbon-based chemistries such as CCl_4 [Tiller and Sameith, 1990]. Gottscho *et al.* [1992] examined inverse RIE lag and concluded they could not see how surface temperature could eliminate RIE lag or cause inverse RIE lag.

The ARDE effects of temperature seen in Figure 8-8 can also be explained in terms of the deposition on the wafer of materials which slows the etching (Figure 8-10). The deposition of such materials is enhanced by lowering the wafer temperature, and thereby enhancing their sticking probability. The deposition occurs on all surfaces of the etching features, but high sticking probabilities on the feature walls can reduce the flux of these species to the bottom of a feature. The observation of the more rapid etching of the small features at lower wafer temperatures is consistent with this "plating out" of the passivation materials on the sidewalls of the high aspect ratio features, and thereby reducing the flux of passivant species to the bottom and increasing the relative etching rate of the smaller features.

Sekine *et al.* [1988],[1993] observed the deposition of $SiCl_x$ and $SiCl_xO_y$ species using mass spectrometry and xray photoelectron spectroscopy (XPS). They observed that deposition was more severe at lower temperatures. In fact, etching was completely suppressed and only deposition was obtained for n^+ polysilicon below -15°C in a $SiCl_4$ magnetron discharge. They observed that deposition precursors were produced in greater quantity in a magnetron than in a standard RIE; this was attributed to the greater ionization efficiency in the magnetron. Further, they observed that deposition was

preferential on polar materials (*e.g.*, Al_2O_3, Si_3N_4, and SiO_2), but would also occur on non-polar materials (*e.g.*, undoped poly, n^+ poly, p^+ poly).

The spacing between lines was measured using SEM photographs as a function of the cathode temperature (Figure 8-23). The effect of deposition on the sidewalls is clearly visible in this plot. A nominal 1.0 μm space is found to be 1.2 μm when etched at 20°C; the same feature is 0.78 μm when etched at -30°C. The decrease in line spacing was due to deposition on the feature sidewalls; this deposition is consistent with the observations of Sekine *et al.* [1988],[1993] for deposition on polar surfaces. SEM photographs for a nominal 1.0 μm space are shown in Figure 8-24. Etching was done at -30°C, 0 Gauss; all other parameters were as listed in Table 8-1. Figure 8-24A shows the post etch feature, with the resist still on. The feature appears to be anisotropic until it is HF dipped (Figure 8-24B) or O_2 ashed. After removing the deposited film, a highly sloped sidewall is seen. The film thickness at the top of the sidewall was approximately 0.2 μm.

The effects of feed gas residence time and cathode temperature on ARDE have both been attributed to deposition from the plasma. In the case of feed gas residence time, increased residence time leads to increased concentrations of depositing species and to more severe RIE lag. In the case of cathode temperature, decreased temperature leads to an increase in the sticking of plasma species and to a reversal from RIE lag to inverse RIE lag. The difference in these two cases is in the depositing species concentrations and in Cl conversion. At long residence times, the concentration of depositing species increases, and the reaction becomes Cl limited as conversion of Cl to $SiCl_4$ approaches 100%. At low temperatures, depositing species stick to the surface more (with essentially

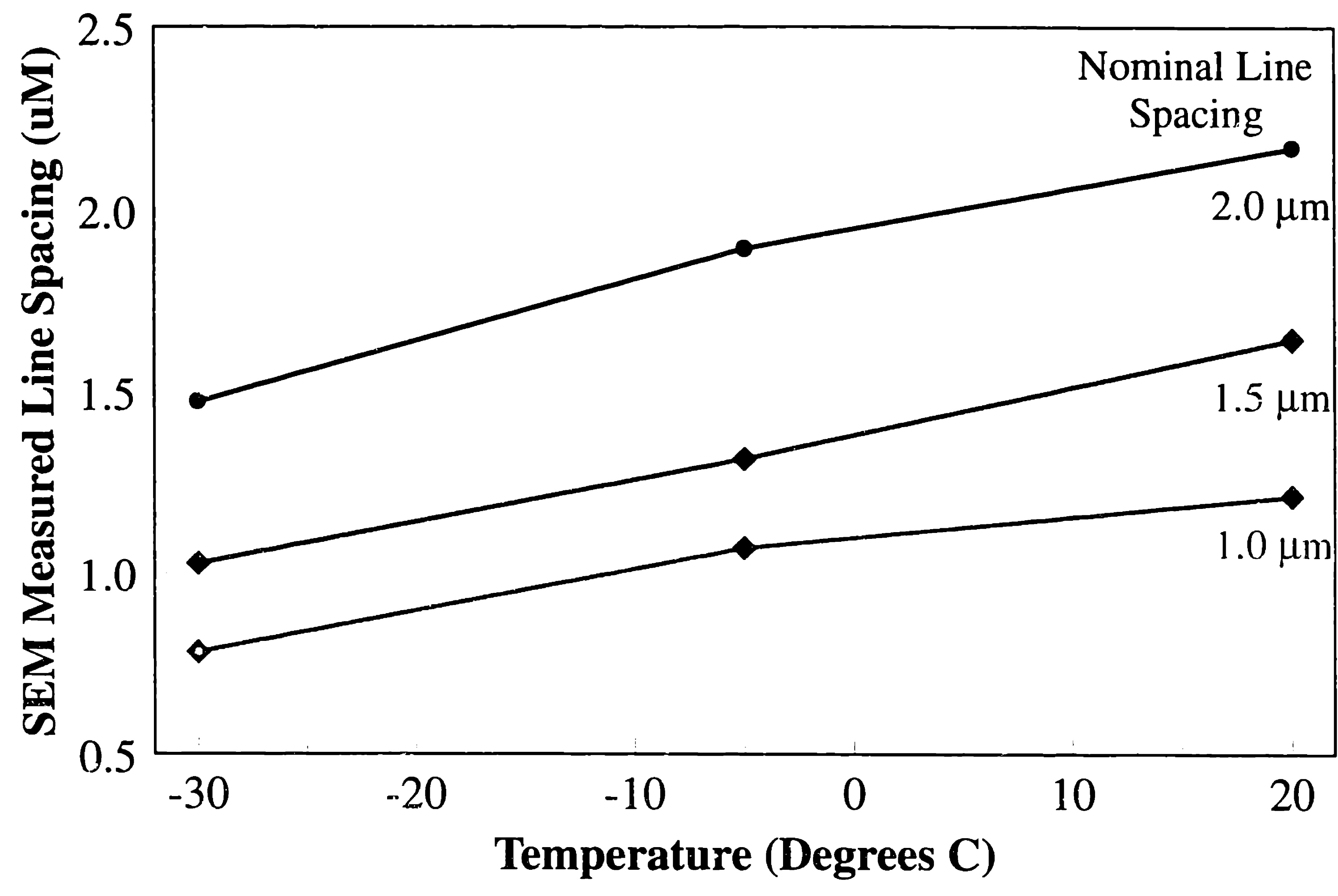


Figure 8-23: SEM Measured Nominal Line Spacings (µm) as a Function of Cathode Temperature (°C) for Nominal Line Spacings of 1.0, 1.5, and 2.0 µm.

A) B)

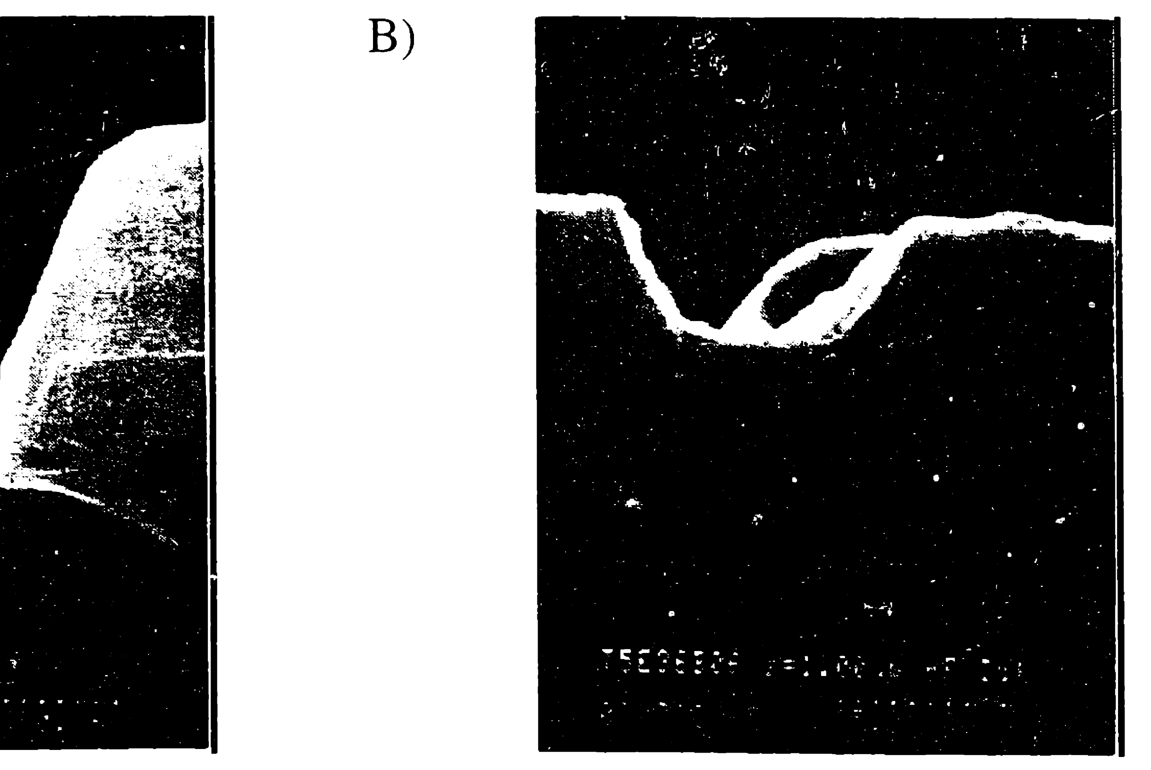


Figure 8-24 : SEM Photographs of 1.0 μm Space Etched at -30°C Showing Sidewall Deposits. A) is Immediately After Etching With the Resist On. B) is After an HF Dip.

the same plasma concentrations), and the conversion of Cl is only around 30%.

8.5.3 Shadowing Effects and Aspect Ratio Dependent Etching

Shadowing caused by the acceptance angle of a feature is thought to have a significant affect on ion, reactant, and polymer transport. inverse RIE lag effects in depositing fluorocarbon systems have been attributed to different acceptance angles for different sized features [Horwitz *et al.*, 1988].

As explained in Section 1.5.1, ions arrive at the wafer surface with a distribution (ion angle distribution (IAD)). The direct ion flux, Φ_D, arriving in a feature can be calculated for a given IAD according to the formula

$$\Phi_D = \int_{\alpha_R}^{\alpha_L} IAD(\theta)\cos(\theta)\,d\theta \qquad \textbf{(8-4)}$$

where α_L and α_R are the limiting view angles to the plasma for a sight in the feature. Measured IADs [Liu *et al.*, 1990] have been fit to a gaussian form [Arnold *et al.*, 1993]. The effect of IAD on ARDE is shown in Figure 8-25, which is a plot of the data of Figure 8-7 along with the computed ERR for values of σ of 2,4,6,8,10,12,14, and 16 degrees. The simulation was conducted for open fields, and for features of 1 μm and 2 μm spacing, with a photoresist angle of 77° and a polysilcion sidewall angle of 90°. The etching rate was computed to be the line average of the etching rate across the bottom of the feature. In general, it is seen here that as σ increases, the amount of RIE lag increases.

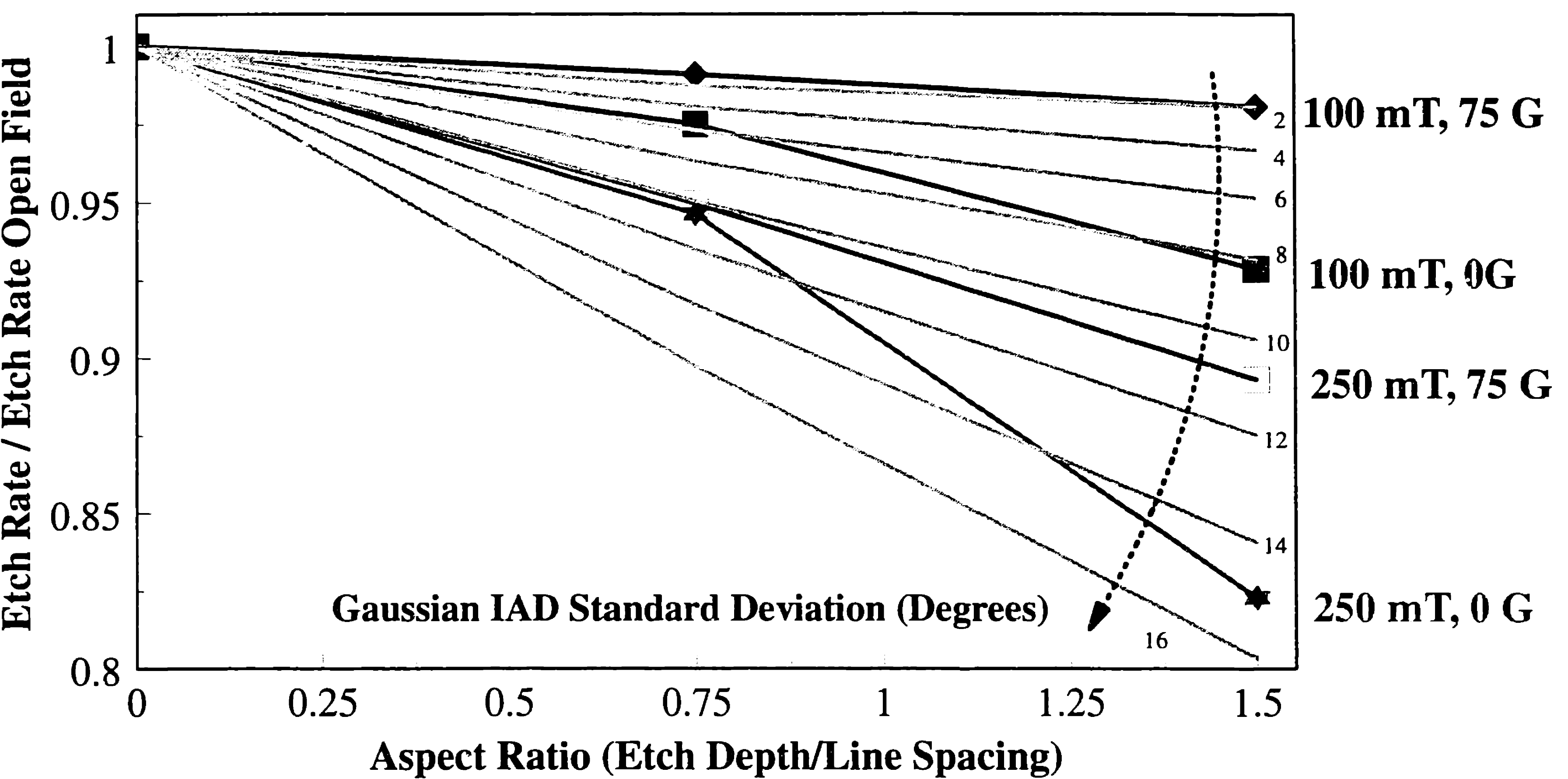


Figure 8-25: Calculated ARDE as a Function of the IAD Standard Deviation. Also Shown is the Data of Figure 8-7.

To understand how the calculated effects of IAD compare to the experimental situations, the IAD needs to be understood. The important parameter for characterizing the shape of the distribution is

$$t_{sh} = \frac{l_{sh}}{\lambda^+} \tag{8-5}$$

where t_{sh} is the dimensionless collisional sheath thickness, l_{sh} is the sheath thickness, and λ^+ is the mean free path for ion-neutral collisions. IADs have been measured for Ar as a function of pressure [Liu *et al.*, 1990]; the IAD standard deviation (σ) and t_{sh} for those measurements are listed in Table 8-4.

Table 8-4

IAD Standard Deviation and Dimensionless Sheath Thickness for Ar

Pressure (mTorr)	σ (Degrees)	t_{sh}
10	4	1
50	6	3.5
500	12	30

Lower pressures (and collisionally thinner sheaths) yield a more directional IAD. The estimated values of t_{sh} for magnetic field strengths of 0 and 75 Gauss are listed in Table 8-5 for a Cl_2 discharge at 100 mT.

Table 8-5

Dimensionless Sheath Thickness for Cl_2

Pressure (mTorr)	t_{sh} B = 0 Gauss	t_{sh} B = 75 Gauss
10	1.4	0.9
100	4.4	2.7
175	5.8	3.6
250	6.9	4.3

The data of Table 8-4 is for Ar, while that of Table 8-5 is for Cl_2. Direct use of the Ar IADs may not be correct; however, the trends should be the same. Table 8-4 shows that as the pressure drops, and the sheath becomes collisionally thinner, the IAD standard deviation decreases. The model shows (Figure 8-25) that as the σ decreases, the effects of ARDE decrease. This is born out by the experimental data. Changing pressure from 250 mTorr to 100 mTorr at both 0 and 75 Gauss decreases the ARDE. The values of σ required to fit the experimental data (15° @ 250 mTorr, 0 Gauss) would require $t_{sh} > 30$ according to the data of Table 8-4. The data of Table 8-5 indicate that at these conditions, t_{sh} is estimated to have a value of 7. Thus, while the numerical values between the two data sets (for two very different chemistries in two very different etchers) may not agree, the trends do agree.

The magnetic field reduces t_{sh} (Table 8-5) by reducing the sheath thickness [Kuypers and Hopman, 1990]. This should result in a lowering of σ, thus resulting in a decreases ARDE effect. The data of Figure 8-7 agree with this trend; increasing the magnetic field decreased the ARDE effects. Again, the predicated values of σ may not

agree with those of Liu *et al.* [1990] for a given value of t_{sh}, but the trends in the data sets do agree.

Besides ion shadowing, there may be neutral shadowing. An extreme case of neutral shadowing has been presented Gottscho *et al.* [1992]. In this limiting case, the etching rate is considered to be neutral limited (following Equation (1-19)); neutrals arrive at the surface with an isotropic distribution, and stick to sidewalls with unity probability (no scattering). In this case, only neutrals with a direct line-of-sight to the feature bottom will arrive there to contribute to etching. Their analysis yielded

$$\frac{R_b}{R_f} = \sin\left[\arctan\left(\frac{1}{2A}\right)\right] \qquad \textbf{(8-6)}$$

where R_b/R_f is the ratio of the etching rate at the bottom center of the trench to the etching rate for a flat region and A is the feature aspect ratio (etch depth / line spacing). This equation predicts a very rapid decrease in etching rate with aspect ratio due to neglecting the flux to the sidewalls (Figure 8-26). At an AR of 1.5, a neutral flux of 30% of that on a plane surface is predicted. This is more extreme than any of our data. Other simulations using experimentally measured kinetics predict a much slower effect of aspect ratio on etching rate [Arnold *et al.*, 1993].

8.6 Conclusions

Aspect ratio dependent etching phenomena were measured for the etching of polysilicon in an HBr/Cl_2 chemistry on an Applied Materials Precision 5000 plasma etcher; spatially resolved optical emission interferometry using a CCD camera was used

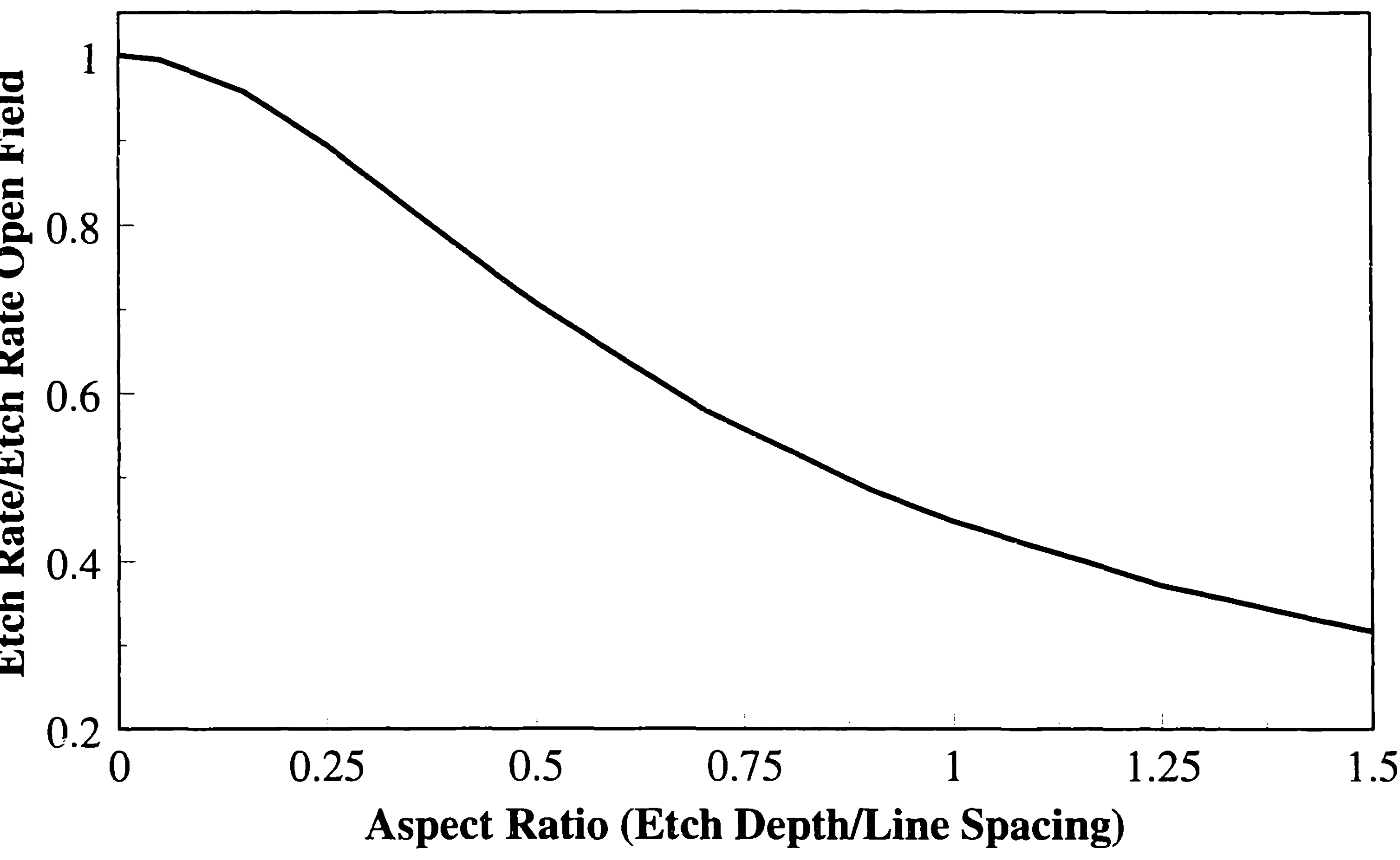


Figure 8-26: ARDE predicted from Neutral Shadowing Model of Gottscho *et al.* [1992].

to measure etching rate *in situ*. A test mask containing gratings of various dimensions was used. We were able to measure variations of etching rate with aspect ratio of only a few percent. Such measurements are very difficult to do using the conventional method of measuring film thicknesses from SEM photographs for a partial etch.

ARDE phenomena were found to be functions of residence time (gas flowrate), pressure, magnetic field strength, temperature, feed gas composition, and the seasoning/history of the etcher.

At high gas flow rates, no ARDE was seen. Decreasing residence time (total gas flow rate) resulted in RIE lag (smaller features etched slower than large features). This was caused by an increase in depositing species caused by the gas phase decomposition of $SiCl_4$ etch products to "sticky" SiCl and $SiCl_2$ radicals. A simple CSTR model for the chemistry that included electron-impact dissociation of Cl_2 and $SiCl_4$ and consumption of Cl to generate $SiCl_4$ followed the experimentally observed optical emission trends with changes in residence time.

RIE lag became more serious with increasing pressure due to ion shadowing caused by the increase in off-normal ion bombardment. A simple model to account for ion shadowing caused by features as a function of the IAD shape followed the experimental trends. Increasing the magnetic field strength resulted in a decrease in RIE lag due to the IAD becoming more directional.

At high temperatures (+20°C), RIE lag was observed. However, as the temperature was lowered to approximately 0°C, the etching rate was independent of feature size. At lower temperatures (-30°C), inverse RIE lag (small features etching faster than large features) was observed. As the temperature was dropped further, the

inverse RIE lag became more severe. In this case, decreasing the temperature resulted in an increase in the sticking coefficient for materials on the wafer surface. In open areas, the materials could stick unimpeeded; they would then have to be etched away, resulting in a decrease in the net etching rate. Inside of narrow features, depositing species would stick on the sidewall, resulting in a decreased flux of these species to the bottom. The next etching rate in these high aspect ratio features would not decrease as much as the open areas or low aspect ratio features due to the lower deposition flux on the feature bottom.

Gas composition had an affect on ARDE. Pure Cl_2 and pure HBr discharges resulted in inverse RIE lag, while 25% / 75% and 50% / 50% (Cl_2 / HBr) discharges yielded etching nearly independent of AR, and 75% / 25% mixtures produced RIE lag.

The condition of the chamber walls also had an affect on ARDE. Cleaning the chamber resulted in a decrease in ARDE for the highest AR features. Also after cleaning, the data were less reproducible. Following heavy use where the chamber again returned to its "seasoned" state, ARDE effects returned to the state they were in before cleaning.

References

Adams, A.C., and C.D. Capio, "Edge profiles in the Plasma Etching of Polycrystalline Silicon," *J. Electrochem. Soc.* **128**(2), 366 (1981).

Adamson, Arthur W., Physical Chemistry of Surfaces(John Wiley & Sons, New York, 1982).

Allen, Kenneth D., Herbert H. Sawin, Michael T. Mocella, and Mary W. Jenkins, "The Plasma Etching of Polysilicon with CF_3Cl/Argon Discharges: I. Parametric Modeling and Impedance Analysis," *J. Electrochem. Soc.* **133**(11), 2315 (1986a).

Allen, Kenneth D., and Herbert H. Sawin, "The Plasma Etching of Polysilicon with CF_3Cl/Argon Discharges: II. Modeling of Ion Bombardment Energy Distribution," *J. Electrochem. Soc.* **133**(11), 2326 (1986b).

Allen, Kenneth D., Herbert H. Sawin, and Akimichi Yokozeki, "The Plasma Etching of Polysilicon with CF_3Cl/Argon Discharges: III. Modeling of Etching Rate and Directionality," *J. Electrochem. Soc.* **133**(11), 2331 (1986c).

Allen, Kenneth D., "Modeling of the Plasma Etching of Polysilicon with Chloro- and Bromo-Trifluoromethane Discharges," Ph.D. Thesis, Massachusetts Institute of Technology, Dept. of Chemical Engineering, (1986d).

Allred, D., S. Jäckel, C. Mazuré, J.J. Barth, H. Cerva, and W. Hösler, "Film Redeposition on Vertical Surfaces During Reactive Ion Etching," *J. Vac. Sci. Technol. B* **7**(3), 505 (1989).

Ameen, Michael, Michael Monfils, Zia Hasan, and Duane Gates, "Silicon Trench Etching Made Easy," *Semicon. Intl.* **11**(10), 122 (1988).

Anders, Hugo, Thin Films in Optics(Focal Press, New York, 1967).

Angell, David and Gottlieb S. Oehrlein, "Grazing Angle Optical Emission Interferometry for End-Point Detection," *Appl. Phys. Lett.* **58**(3), 240 (1991).

Arikado, T., M. Sekine, H. Okano, and Y. Horiike, "Influence of the Magnetic Field on the Gate Oxide Breakdown Phenomenon During Reactive Ion Etching," *Proc. 8th Int. Symp. Plasma Chem.*, 979 (1987).

Arnold, John C., and Herbert H. Sawin, "Charging of Pattern Features During Plasma Etching," *J. Appl. Phys.*, **70**(10), 5314 (1991).

Arnold, John C., David C. Gray, and Herbert H. Sawin, "Influence of Reactant Transport on Fluorine Reactive Ion Etching of Deep Trenches in Silicon," *J. Vac. Sci. Technol. B* **11**(6), 2071 (1993).

Arnold, John C., Ph.D. Thesis, Massachusetts Institute of Technology, Dept. of Chemical Engineering, (1994).

ASEPEN - "Algorithmic Simulator for Etching Profile Evolution", du Point Electronics, Wilmington, DE (1990).

Aydil, Eray S., and Demetre J. Economou, "Modeling of Plasma Etching Reactors Including Wafer Heating Effects," *J. Electrochem. Soc.* **140**(4), 1471 (1993).

Barker, R.A., T.M. Mayer, and W.C. Pearson, "Surface Studies of a Mass Balance Model for Ar^+ Ion-Assisted Cl_2 Etching of Si," *J. Vac. Sci. Technol. B* **1**(1), 37 (1983).

Barrett, John H., "Monte Carlo Channeling Calculations," *Phys. Rev. B.*, **3**(5), 1527 (1971).

Bestwick, Tim D., and Gottlieb S. Oehrlein, "Reactive Ion Etching of Silicon Using Bromine Containing Plasmas," *J. Vac. Sci. Technol. A* **8**(3), 1696 (1990a).

Bestwick, Tim D., Gottlieb S. Oehrlein, and David Angell, "Cryogenic Reactive Ion Etching of Silicon in SF_6," *Appl. Phys. Lett.* **57**(5), 431 (1990b).

Bestwick, Tim D., Gottlieb S. Oehrlein, Ying Zhang, and Gerrit M. W. Kroesen, "Reactive Ion Etching of SiGe Alloys Using HBr," *Appl. Phys. Lett.* **59**(3), 336 (1991).

Bhagat, Phiroz, "An Introduction to Neural Nets," *Chem. Eng. Prog.* **86**(8), 55 (1990).

Biersack, J.P., and L.G. Haggmark, "A Monte Carlo Computer Program for the Transport of Energetic Ions in Amorphous Targets," *Nucl. Instrum. Meth.*, **174**, 257 (1980).

Bird, R. Byron, Warren E. Stewart and Edwin N. Lightfoot, Transport Phenomena(John Wiley & Sons, New York, 1960).

Boning, Duane, Jonathan Claman, Ka Shun Wong, Herb Sawin, and Timothy J. Dalton, "Plasma Etching Endpoint via Interferometric Imaging," ACC Invited Session on Modeling and Control in Microelectronics (1994).

Bower, Douglas H., "Planar Plasma Etching of Polysilicon Using CCl_4 and NF_3," *J. Electrochem. Soc.* **192**(4), 795 (1982).

Box, G.E.P., W.G. Hunter, and J.S. Hunter, Statistics for Experimenters (John Wiley and Sons, New York, 1978).

Braga, E.S., G.F. Mendes, J. Frejlich, and A.P. Mammana, "Optical Monitoring of the End Point in Thin Film Plasma Etching," *Thin Solid Films* **109**, 363 (1983).

Brown, H.L, G.B. Bunyard, and K.C. Lin, "Applications of Mass Spectrometers to Plasma Process Monitoring & Control," *Solid. State. Technol.* **21**(7), 35 (1978).

Busta, H.H., R.E. Lajos, and D.A. Kiewit, "Plasma Etch Monitoring with Laser Interferometry," *Solid State Technol.* **22**(2), 61 (1979).

Butterbaugh, J.W., L.D. Baston, and H.H. Sawin, "Measurement and Analysis of Radio Frequency Glow Discharge Electrical Impedance and Network Power Loss," *J. Vac. Sci. Technol. A* **8**(2), 916 (1990).

Butterbaugh, J.W., and H.H. Sawin, "Erratum: Measurement and Analysis of Radio Frequency Glow Discharge Electrical Impedance and Network Power Loss," *J. Vac. Sci. Technol. A* **10**(3), 578 (1992).

Chang, Yan-Tyng, Ching-Jiang Hwang, and Tzu-Min Su, "Measurement of the Diffusion Coefficient of Atomic Chlorine in Molecular Chlorine," *Chem. Phys. Lett.* **114**(1), 92 (1985).

Chapman, Brian, Glow Discharge Processing(John Wiley and Sons, New York, 1980).

Chin, Daeje, Sang H. Dhong, and Glenn J. Long, "Structural Effects on a Submicron Trench Process," *J. Electrochem. Soc.* **132**(7), 1705 (1985).

Coburn, J.W., and Harold F. Winters, "Ion- and Electron-Assisted Gas-Surface Chemistry - An Important Effect in Plasma Etching," *J. Appl. Phys.* **50**(5), 3189(1979).

Coburn, J.W., and M. Chen, "Optical Emission Spectroscopy of Reactive Plasmas: A method for Correlating Emission Intensities to Reactive Particle Density," *J. Appl. Phys.* **51**(6), 3134 (1980).

Coburn, J.W., "Summary Abstract: Diagnostics in Plasma Processing," *J. Vac. Sci. Technol. A* **4**(3), 1830 (1986).

Coburn, J.W., and Harold F. Winters, "Conductance Considerations in the Reactive Ion Etching of High Aspect Ratio Features," *Appl. Phys. Lett.* **55**(26), 2730 (1989).

Collot, P., T. Diallo, and J. Canteloup, "Dry-etch Monitoring of III-V Heterostructures using Laser Reflectometry and Optical Emission Spectroscopy," *J. Vac. Sci. Technol. B* **9**(5), 2497 (1991).

Conner, William T., *Personal Communication* (1991).

Cook, J.M., D.E. Ibbotson, and D.L. Flamm, "Application of Low-Pressure Radio Frequency Discharge Source to Polysilicon Gate Etching," *J. Vac. Sci. Technol. B* **8**(1), 1 (1990).

Cooper, C.B. III., M.E. Day, C. Yuen, and M. Salimian, "Reactive Ion Etching of Through-the-Wafer Via Connections for Contacts to GaAs FET's," *J. Electrochem. Soc.* **134**(10), 2533 (1987).

Curtis, B.J., and H.R. Brunner, "The Reactive Ion Etching of Tantalum Silicide/Polysilicon BiLayers in Silicon Tetrachloride and Hydrogen Chloide," *J. Electrochem. Soc.* **136**(5), 1463 (1989).

Dalton, Jeff, and Atul Deshmane, "Artificial Neural Networks," *IEEE Potentials*, **10**(2), 33 (1991).

Dalton, T.J., J.C. Arnold, H.H. Sawin, D. Corliss, and S. Swan, "Microtrench Formation in Polysilicon Plasma Etching Over Thin Gate Oxide," *J. Electrochem. Soc.* **140**(8), 2395 (1993).

Danner, D.A., M. Dalvie, and D. W.Hess, "Plasma Etching of Aluminum:A Comparison of Chlorinated Etchants," *J. Electrochem. Soc.* **134**(3), 669 (1987).

Davies, John T., Thomas Metz, Richard N. Savage and Horace Simmons, "Real-time, *in situ* Measurement of File Thickness and Uniformity During Plasma Ashing of Photoresist," *Proc. SPIE*, **1392**, 551 (1990).

Deen, William M., and Robert A. Brown, "Analysis of Transport Phenomena," M.I.T. 10.50 Course Notes, Massachusetts Institute of Technology, Cambridge, MA (1988).

Degenkolb, E.O. C.J. Mogab, M.R. Goldrick, and J.E. Griffiths, "Spectroscopic Study of Radiofrequency Oxygen Plasma Stripping of Negative Photoresists. I. Ultraviolet Spectrum," *Appl. Spectrosc.* **30**(5), 520 (1976).

Degenkolb, E.O., and J.E. Griffiths, "Simple Optical Devices for Detection of Radiofrequency Oxygen Plasma Stripping of Photoresists," *Appl. Spectrosc.* **31**(1), 40 (1977).

Dolins, Steven B., Aditya Srivastava, and Bruce E. Flinchbaugh, "Monitoring and Diagnosis of Plasma Etching Processes," *IEEE Trans. Semi. Manuf.* **1**(1), 23 (1988).

Donnelly, Vincent M., and Daniel L. Flamm, "Effects of Frequency on Optical Emission, Electrical, Ion, and Etching Characteristics of a Radio Frequency Chlorine Plasma," *J. Appl. Phys.* **58**(6), 2135 (1985).

Donnelly, V.M., D.E. Ibbotson, and C.-P. Chang, "Interferometric Thermometry Measurements of Silicon Wafer Temperatures During Plasma Processing," *J. Vac. Sci. Technol. A* **10**(4), 1060 (1992).

Donnelly, V.M., "Real-Time Determination of the Direction of Wafer Temperature Change by Spatially Resolved Infrared Laser Interferometric Thermometry," *J. Vac. Sci. Technol. A* **11**(5), 2393 (1993).

Ducommun, J.P., M. Cantagrel, and M. Moulin, "Evolution of Well-Defined Surface Countour Submitted to Ion Bombardment: Computer Simulation and Experimental Investigation," *J. Mater. Sci.* **10**(1), 52 (1975).

Economou, Demetre J., and Richard Alkire, "Effect of Potential Field on Ion Deflection and Shape Evolution of Trenches During Plasma-Assisted Etching," *J. Electrochem. Soc.* **135**(4), 941 (1988).

Economou, D., E. Aydil and G. Barna, "*In situ* Monitoring of Etching Uniformity in Plasma Reactors," *Solid State Tech.*, **34**(4), 107 (1991).

Edwards, David F., "Silcion (Si)", in Handbook of Optical Constants of Solids, Edward D. Palik, Ed., (Academic Press, Orlando FL, 1985).

Egerton, E.J., A. Nef, W. Millikin, W. Cook, and D. Baril, "Positive Wafer Temperature Control to Increase Dry Etch Throughput and Yield," *Solid State Technol.* **25**(8), 84 (1982).

El-Masry, Ahmed M., F-O. Fong, J.C. Wolfe, and John N. Randall, "Magnetically Enhanced Reactive Ion Etching of Silicon in Bromine Plasmas," *J. Vac. Sci. Technol. B* **6**(1), 257 (1988).

Ephrath, L.M., and R.S. Bennett, "Directional and Selective Etching of Polysilicon and Polycide," *Proc. Electrochem. Soc.* 82-7, 108 (1982).

Esselman, Nicholas, Applied Materials Hudson Office *Personal Communication*, (1991).

Fahien, Ray W , Fundamentals of Transport Phenomena (McGraw-Hill, New York, 1983), p.343.

Fang, Sychyi, and James P. McVittie, "Charging Damage to Gate Oxides in an O_2 Magnetron Plasma," *J. Appl. Phys.* **72**(10), 4865 (1992).

Fang, Sychyi, and James P. McVittie, "Model for Oxide Damage from Gate Charging During Magnetron Etching," *Appl. Phys. Lett.* **62**(13), 1507 (1993).

Fedynyshyn, Theodore H., Gregory W. Grynkewich, Bomy A. Chen, and T.P. Ma., "The effect of Metal Masks on the Plasma Etch Rate of Silicon," *J. Electrochem. Soc.* **136**(6), 1799 (1989).

Flamm, Daniel L., and Vincent M. Donnelly, "The Design of Plasma Etchants," *Plasma Chem. Plasma Proc.* **1**(4), 317 (1981).

Florio, J.V., and W.D. Robertson, "Chlorine Reactions on the SI(111) Surface," *Surf. Sci.*, **18**, 398(1969).

Fonash, Stephen J., "An Overview of Dry Etching Damage and Contamination Effects," *J. Electrochem. Soc.* **137**(12), 3885 (1990).

Frank, W.E., and T. Chabert, "Dry Etching of Single-Crystal Silicon Trench in Hydrogen Iodine Containing Plasmas," *J. Electrochem. Soc.* **140**(2), 490 (1993).

Fujiwara, Nobuo, Teruo Shibano, Kyusaku Nishioka, and Tadao Kato, "Cold and Low-Energy Ion Etching (COLLIE)," *Jap. J. Appl. Phys.* **28**(10), 2147 (1989).

Fujiwara, Nobuo, Hisahuru Sawai, Masahiro Yoneda, Kyusaku Nishioka, and Haruhiko Abe, "ECR Plasma Etching With Heavy Halogen Ions," *Jap. J. Appl. Phys.* **29**(10), 2223 (1990).

Gabriel, Calvin T., "Gate Oxide Damage From Polysilicon Etching," *J. Vac. Sci. Technol. B.* **9**(2), 370 (1991).

Gabriel, Calvin T., and James P. McVittie, "How Plasma Etching Damages Thin Gate Oxides," *Solid State Tech.*, **35**(6), 81 (1992).

Geankoplis, Christie J., Mass Transport Phenomena (Ohio State University Bookstore, Ohio, 1984).

Giapis, Konstantino P., Geoffrey R. Scheller, Richard A. Gottscho, William S. Hobson & Yong H. Lee, "Microscopic and Macroscopic Uniformity Control in Plasma Etching," *Appl. Phys. Lett.* **57**(10), 983 (1990).

Glöersen, Per G., "Ion-Beam Milling," *J. Vac. Sci. Technol.* **12**(1), 28 (1975).

Godyak, V.A., and R.B. Piejak, "Paradoxical Spatial Distribution of the Electron Temperature in a Low Pressure RF Discharge," *Appl. Phys. Lett.* **62**(23), 3137 (1993).

Gogolides, Evangelos, and Herbert H. Sawin, "N^+-Polysilicon Etching in CCl_4/He Discharges:Characterization and Modeling," *J. Electrochem. Soc.* **136**(4), 1147 (1989).

Gottscho, Richard A., and Terry A. Miller, "Optical Techniques in Plasma Diagnostics," *Pure and Appl. Chem.*, **56**(2), 189 (1984).

Gottscho, Richard A., C.W. Jurgensen, and D.J. Vitkavage, "Microscopic Uniformity in Plasma Etching," *J. Vac. Sci. Technol.* **B** 10(5), 2133 (1992).

Graves, David, "Plasma Processing in Microelectronics Manufacturing," *AICHE Journal*, **35**(1), 1 (1989).

Gray, David C., "Beam Simulation Studies of Plasma-Surface Interactions in Fluorocarbon Etching of Si and SiO_2," Ph.D. Thesis, Massachusetts Institute of Technology, Dept. of Chemical Engineering, (1992).

Gray, D.C., I. Tepermeister, and Herbert H. Sawin, "Phenomenological Modeling of Ion-Enhanced Surface Kinetics in Fluorine-Based Plasma Etching," *J. Vac. Sci. Technol. B* **11**(4), 1243 (1993).

Greene, W.M., J.B. Kruger, and G. Kooi, "Magnetron Etching of Polysilicon:Electrical Damage," *J. Vac. Sci. Technol. B.* **9**(2), 366 (1991).

Griffiths, J.E., and E.O. Degenkolb, "Spectroscopic Study of Radiofrequency Oxygen Plasma Stripping of Negative Photoresists. II. Visible Spectrum," *Appl. Spectrosc.* **31**(2), 134(1977).

Grynkewich, Gregory W., Theodore H. Fedynyshyn, and Richard H. Dumas, "The Effect of Aluminum Masks on the Plasma Etch Rates of Polysilicon and Silicon Nitride," *J. Vac. Sci. Technol. B* **8**(1), 5 (1990).

Guite, D.R., A.I. Spiers, and A.C. Sharp, "A Cl_2/SF_6 Polysilicon Plasma Etch Process for Non-Planar Topologies," *Proc. Electrochem. Soc.* 85-1, 545 (1985).

Harshbarger, W.R., R.A. Porter, and P. Norton, "Optical Detector to Monitor Plasma Etching," *J. Elec. Mater.* **7**(3), 429 (1978).

Harshbarger, William R., "Plasma Diagnostics and End-Point Detection," in VLSI Electronics Microstructure Science Vol. 8:Plasma Processing for VLSI(Academic Press, New York, 1984), 411.

Hasted, J.B., "Ion-Atom Charge Transfer Collisions at Low Energies", in Advances in Atomic and Molecular Physics, Vol. 15, D.R. Bates and B. Bederson, Editors, (Academic Press, New York, 1979), p. 205.

Hayes, T.R., M.A. Dreisbach, P.M. Thomas, W.C. Dautremont-Smith, and L.A. Heimbrook, "Reactive Ion Etching of InP using CH_4/H_2 Mixtures: Mechanisms of Etching and Anisotropy," *J. Vac. Sci. Technol. B* **7**(5), 1130(1989).

Hecht, Eugene, and Alfred Zajac, OPTICS (Addison Wesley, Reading MA., 1979).

Heinrich, F., H.-P. Still, and H.-C. Scheer, "New and Simple Optical Method to *in situ* etch rate determination and endpoint detection," *Appl. Phys. Lett.* **55**(14), 1474 (1989).

Henck, Steven A., "*In situ* Real-Time Ellipsometry for Film Thickness Measurement and Control," *J. Vac. Sci. Technol. A* **10**(4), 934 (1992).

Henck, Steven A., Walter M. Duncan, Lee M. Lowenstein, and Stephanie Watts Butler, "*In Situ* Spectral Ellipsometry for Real-Time Thickness Measurement: Etching Multilayer Stacks," *J. Vac. Sci. Technol. A* **11**(4), 1179 (1993).

Hills, Graham W., "Plasma Charge-Up Effects in 0.5 μm Device Fabriaction," *Applied Materials Spring Technical Seminars*, (1993) Unpublished.

Hirboe, Kado, Ko-ichirou Kawamura, and Kazuo Nojiri, "Formation of Deep Holes in Silicone by Reactive Ion Etching," *J. Vac. Sci. Technol. B* **5**(2), 594 (1987).

Hitchman, M. L., and V. Eichenberger, "A Simple Method of End-Point Determintion for Plasma Etching," *J. Vac. Sci. Technol.* **17**(6), 1378 (1980).

Horiike, Y., T. Arikado, K. Horioka, M. Sekine, H. Nishino, N. Hayasaka, and H. Okano, "Recent Progress in Dry Etching Technology for Advanced VLSI Devices," *Vide Couches Minces*, **237**(suppl.), 173 (1987).

Horioka, K., T. Arikado, M. Sekine, H. Okano, and Y. Horiike, "Highly Selective and Directional Etching of Phosphorus Doped Polycrystalline Silicon with Tri-Level Resist Mask Employing Magnetron Plasma," *Proc. 1988 Symp. VLSI Technol.*, 81 (1988).

Horwitz, Chris M., S. Boronkay, M. Gross, and K. Davies, "Hollow Cathode Etching and Deposition," *J. Vac. Sci. Technol. A* **6**(3), 1837 (1988).

Hou, M., and M.T. Robinson, "The Conditions for Total Reflection of Low-Energy Atoms from Crystal Surfaces," *Appl. Phys.*, **17**, 371 (1978).

Hwang, Ching-Jiang, and Tzu-Min Su, "Diffusion Coefficients for Atomic Chlorine in Inert Molecules," *J. Chem. Phys.*, **91**, 2351 (1987).

Hwang, Ching-Jiang, and Tzu-Min Su, "Measurement of the Diffusion Coefficients of Atomic Bromine in Inert Gases," *Chem. Phys. Lett.* **164**(4), 383 (1989).

Hwang, Ching-Jiang, R.-C. Jiang, and Tzu-Min Su, "Measurement of the Diffusion Coefficients for Atomic Bromine in Rare Gases," *J. Chem. Phys.*, **91(3)**, 1626 (1989).

Ibbotson, Dale E., Daniel L. Flamm, and Vincent M. Donnelly, "Crystallographic Etching of GaAs with Bromine and Chlorine Plasmas," *J. Appl. Phys.* **54**(10), 5974 (1983).

Jackson, Scott C., and Timothy J. Dalton, "A Profile Evolution Model With Redeposition," *Proc. SPIE.*, **1185**, 225 (1989).

Janes, Joachim, and Wolfgang Pilz, "Effect of Ion Angular Distributions on Microloading in Oxygen Reactive-Ion Etching of Submicron Polymer Trenches," *J. Appl. Phys.* **74**(1), 649 (1993).

Jenkins, M.W., M.T. Mocella, K.D. Allen, and H.H. Sawin, "The Modeling of Plasma Etching Processes Using Response Surface Methodology," *Solid State Technol.* **29**(4), 175 (1986).

Johnson, David, "Optical Methods Detect End Point in Plasma Etching," *Ind. Res. Dev.* **22**(10), 181 (1980).

Kao, Alan S., and Harvey G. Stenger, Jr., "Analysis of Nonuniformities in the Plasma Etching of Silicon," *J. Electrochem. Soc.* **137**(3), 954 (1990).

Karulkar, Pramod C., and Mark A. Wirzbicki, "Plasma Etching of Ion Implanted Polysilicon," *J. Electrochem. Soc.* **136**(9), 2716 (1989).

Kawamoto, Yoshifumi and Norikazu Hashimoto, "Plasma Etching Monitor by Electric Probe," *Jap. J. Appl. Phys. Suppl.* **18**(1), 277 (1979).

Kawamoto, Yoshifumi, "Measurement of Charge-Up Voltage in Glow Discharge Using MNOS Capacitor," *Ext. Abs. 17th Conf. Solid State Dev. and Matl.*, 333 (1985).

Kay, Eric, "Reactive Ion Etching and Related Polymerization Process," in Methods and Materials in Microelectronics Technology, J. Bargon, Ed.(Plenum, New York, 1984), p. 243.

Kelly, Roger, and Orlando Auciello, "On the Origin of Pyramids and Cones on Ion-Bombarded Cipper Surfaces", *Surf. Sci.*, **100**, 135 (1980).

Kimizuka, M., and K. Hirata, "Pattern Profile Control of Polysilicon Plasma Etching," *J. Vac. Sci. Technol B* **3**(1), 16 (1985).

Kimizuka, Masakatsu, Yoshio Watanabe, and Yoshiharu Ozaki, "Pattern profile Control in Magnetron Reactive Ion Etching of poly-Si," *J. Vac. Sci. Technol. B.* **10**(5), 2192 (1992).

Kiss, L.D.B., "Study of Plasma Chemical Kinetics by Modulated Power Relaxation," Ph.D. Thesis, Massachusetts Institute of Technology, Dept. of Chemical Engineering, (1991).

Kiss, L.D.B., and H.H. Sawin, "Evaluation of CF_4 Plasma Chemistry by Power Modulation," *Plasma Chem. Plasma Proc.* **12**(4), 523 (1992).

Kleinknecht, H.P., and H. Meier, "Optical Monitoring of the Etching of SiO_2 and Si_3N_4 on Si by the Use of Grating Test Patterns," *J. Electrochem. Soc.* **125**(5), 798 (1978).

Krukar, Richard, Avi Kornbilt, Linda A. Clark, Joseph Kruskal, Diane Lambert, Edward A. Reitman, and Richard A. Gottscho, "Reactive Ion Etching Profile and Depth Characterization Using Statistical and Neural Network Analysis of Light Scattering Data," *J. Appl. Phys.* **74**(6), 3698 (1993).

Kushner, M.J., "Distribution of Ion Energies Incident on Electrodes in Capacitively Coupled RF Discharges," *J. Appl. Phys.* **58**(11), 4024 (1985).

Kuypers, A.D., and H.J. Hopman, "Measurement of Ion Energy Distributions at the Powered RF Electrode in a Variable MagneticField," *J. Appl. Phys.* **67**(3), 1229 (1990).

Lee, Robert E., "Microfabrication by Ion-Beam Etching," *J. Vac. Sci. Technol.* **16**(2), 164 (1979).

Lee, Young H., and Zhen H. Zhou, "Feature-Size Dependence of Etch Rate in Reactive Ion Etching," *J. Electrochem. Soc.* **138**(8), 2439 (1991).

Lehmann, H.W., L. Krausbauer, and R. Widmer, "Redeposition - A Serios Problem in rf Sputter Etching of Structures with Micrometer Dimensions," *J. Vac. Sci. Technol.* **14**(1), 281 (1977).

Lii, Y.-J. Tom, and Jacob Jorné, "Redeposition During Deep Trench Etching," *J. Electrochem. Soc.* **137**(9), 2837(1990).

Light, R.W., and H.B. Bell, "Profile Control of Polysilicon Lines with an SF_6/O_2 Plasma Etch Process," *J. Electrochem. Soc.* **130**(7), 1567 (1983).

Liu, J., G.L. Huppert, and H.H. Sawin, "Ion Bombardment in rf Plasmas", *J. Appl. Phys.* **68**(8), 3916 (1990).

Liu, Joanne, "Scaling Relationships for Power Deposition and Ion Bombardment in Radio-Frequency Plasmas," Ph.D. Thesis, Massachusetts Institute of Technology, Dept. of Chemical Engineering, (1993).

Maa, J.-S., H. Gossenberger, and F. DiGeronimo, "Anisotropic Etching of Polysilicon in a Single-Wafer Aluminum Etch Reactor," *J. Vac. Sci. Technol. B* **9**(3), 1596 (1991).

Madix, Robert J., and James A. Schwarz, "Chemical Relaxation Molecular Beam Studies of Reactive Gas-Solid Scattering. I:Reaction of Silicon and Germanium with Molecular Chlorine," *Surf. Sci.* **24**, 264 (1971).

Maissel, Leon I., and Reinhard Gland, ed., Handbook of Thin Film Technology (McGraw-Hill, New York, 1970).

Manenschijn, A., and W.J. Goedheer, "Angular Ion and Neutral Energy Distributions in a Collisional rf Sheath," *J. Appl. Phys.* **69**(5), 2923 (1991).

Marcoux, Paul J., and Pang Dow Foo, "Methods of End Point Detection for Plasma Etching," *Solid State Tech.*, **24**(4), 115 (1981).

Mayer, T.M., and R.A. Baker, "Simulation of Plasma-Assisted Etching Processes by Ion-Beam Techniques," *J. Vac. Sci. Technol.* **21**(3), 757 (1982).

Mayer, T.M., M.S. Ameen, E.L. Barish, T. Mizutani, and D.J. Vitkavage, "Summary Abstract: Ion-Enhanced Processes in Etching of Silicon," *J. Vac. Sci. Technol. B*, **3**(5), 1373 (1985).

McNevin, S.C., "Radio Frequency Plasma Etching of Si/SiO_2 by Cl_2/O_2: Improvements Resulting From the Time Modulation of the Processing Gases," *J. Vac. Sci. Technol. B* **8**(6), 1185 (1990).

McNevin, S.C., "Radio Frequency Plasma Etching of Si/SiO_2 by Cl_2/O_2: Improvements Resulting From the Time Modulation of the Processing Gases," *J. Vac. Sci. Technol. A* **9**(3), 816 (1991).

McNevin, S.C., D.J. Vitkavage, and C.N. Bredbenner, "Bias Voltage Diagnostics During Oxide Etch in Drytek 384T," *J. Vac. Sci. Technol. A* **11**(4), 1142 (1993).

Mieth, M., and A. Barker, "Anisotropic Etching of Polysilicon using SF_6 and $CFCl_3$," *J. Vac. Sci. Technol. A* **1**(2), 629 (1983).

Mitchner, M., and Charles H. Kruger, Partially Ionized Gases(John Wiley and Sons, New York, 1973).

Mocella, Michael T., James A. Bondur, and Terry R. Turner, "Etch Process Characterization Using Neural Network Methodology: A Case Study," *Proc. SPIE.*, **1594**, 232(1991).

Mocella, Michael T., *Personal Communication*, (1991b).

Mogab, C.J., and H.J. Levinstein, "Anisotropic Plasma Etching of Polysilicon," *J. Vac. Sci. Technol.* **17**(3), 721 (1980).

Moreno-Marin, J.C., J.A. Valles-Abarca, and A. Gras-Marti, "Secondary Effects in Ion Milling," *J. Vac. Sci. Technol B* **4**, 322 (1986).

Müller-Markgraf, W., and M.J. Rossi, "The Interaction of $Cl(^2P_{3/2})$ and $Cl(^2P_{1/2})$ with n-Si(100): Spontaneous Etching," *J. Vav. Sci. Technol. A* **9**(2), 217 (1991).

Nagy, A.G., "Radial Etch Rate Nonuniformity in Reactive Ion Etching," *J. Electrochem. Soc.* **131**(8), 1871 (1984).

Nakamura, Moritaka, Katsuhiko Iizuka, and Hiroshi Yano, "Variable Profile poly-Si etching with Low Temperature RIE and HBr Gas," *1988 Dry Process Symposium*, 58 (1988).

Nakamura, Moritaka, Katsuhiko Iizuka, and Hiroshi Yano, "Very High Selective n^+ poly-Si RIE with Carbon Elimination," *Jap. J. Appl. Phys.* **28**(10), 2142 (1989).

Nguyen, San Van, Greg Chrisman, Dave Dobuzinsky, and Dave Harmon, "Magnetically Enhanced Reactive Ion Etching of Poly Gate Electrodes Smaller than 0.5 μm," *Solid State Technol.* **33**(10), 73 (1990).

Nguyen, Son Van, Dave Dobuzinsky, Scott R. Stiffler, and Greg Chrisman, "Substrate Trenching Mechanism During Plasma and Magnetically Enhanced Polysilicon Etching," J. Electrochem. Soc. 138(4), 1112 (1991).

NNAPER - "Neural Network Analysis of Process Evaluation Responses", du Pont Electronics, Semiconductor Materials Division, Wilmington, DE, V3.0, (1993).

Nojiri, Kazuo, and Kazuyuki Tsunokuni, "Study of Gate Oxide Breakdown Caused by Charge Buildup During Dry Etching," *J. Vac. Sci. Technol. B* **11**(5), 1819 (1993).

Nomura, Y., Y. Morishita, S. Goto, Y. Katayama, and T. Isu, "Surface Diffusion Length of Ga adatoms on (111) B Surfaces during Molecular Beam Epitaxy," *Appl. Phys. Lett.* **4**(9), 1123 (1994).

Oehrlein, Gottlieb S., R.M. Tromp, Y.H. Lee, and E.J. Petrillo, "Study of Silicon Contamination and Near-Surface Damage Caused by CF_4/H_2 Reactive Ion Etching," *Appl. Phys. Lett.* **45**(4), 420 (1984).

Oehrlein, Gottlieb S., S.W. Robey, J.L. Lindström, K.K. Chan, M.A. Jaso, and G.J. Scilla, "Surface Modifications of Electronics Materials Induced by Plasma Etching," *J. Electrochem. Soc.*, **136**(7), 2050 (1989).

Oehrlein, Gottlieb S., J.F. Rembetski, and E.H. Payne, "Study of Sidewall Passivation and Microscopic Silicon Roughness Phenomena in Chlorine-Based Reactive Ion Etching of Silicon Surfaces," *J. Vac. Sci. Technol. B.* **8**(6), 1199 (1990).

Oehrlein, Gottlieb S., "Effects of Ion Bombardment in Plasma Etching on the Fluorinated Silicon Surface Layer: Real-Time and Postplasma Surface Studies," *J. Vac. Sci. Technol. A* **11**(1), 34 (1993).

Ogryzlo, Elmer A., Daniel L. Flamm, Dale E. Ibbotson, and John A. Mucha, "The Etching of Doped Polycrystalline Silicon by Molecular Chlorine," *J. Appl. Phys.* **64**(11), 6510 (1988).

Ogryzlo, Elmer A., Dale E. Ibbotson, Daniel L. Flamm, and John A. Mucha, "Doping and Crystallographic Effects in the Cl-Atom Etching of Silicon," *J. Appl. Phys.* **67**(5), 3115 (1990).

Ohki, Shigehisa, Masatoshi Oda, Hideo Akiya, and Toshitaka Shibata, "Cavernous Undercuts in Reactive Ion Etched Submicron-Wide Deep Trenches," *J. Vac. Sci. Technol. B* **5**(6), 1611 (1987).

Oostra, D.J., R.P. Van Ingen, A. Haring, A.E. de Vries, and G.N.A. Van Veen, "Near Threshold Sputtering of Si and SO_2 in a Cl^2 Environment," *Appl. Phys Lett.* **50**(21), 1506 (1987).

Owens, Aaron J., and Michael T. Mocella, "An Experimental Design Advisor And Neural Network Analysis Package," *IEEE Intl. Workshop on Artificial Neural Networks*, Grenada Spain, (1991).

Paranjpe, Ajit P., James P. McVittie, and Sidney A. Self, "A Tuned Langmuir Probe for Measurements in rf Glow Discharges," *J. Appl. Phys.* **67**(11), 6718 (1990).

Parrillo, L.C., "VLSI Process Integration" in VLSI Technology,S.M. Sze, ed.(McGraw-Hill, New York, 1983), 446.

Patel, V. B. Singh, and J.H. Thomas III, "Reactive Ion Etching End-Point Determination by Plasma Impedance Monitoring," *Appl. Phys. Lett.* **61**(16), 1912 (1992).

Patel, V., M. Patel, S. Ayyagari, W.F. Kosonocky, D. Misra, and B. Singh, "Wafer Temperature Measurements and End-Point Detection During Plasma Etching by Thermal Imaging," *Appl. Phys. Lett.* **59**(11), 1299 (1991).

Pearse, Reginald W.B., and Alfred G. Gaydon, The Identification of Molecular Spectra (Chapman and Hall, London, 1984).

Peng, Shiesen, Devereaux C. Chen, and E.D. Liu, "Optimization of Submicron Polysilicon Etching and the Effects of Organic and Inorganic Masks, and Their Aspect Ratios," *J. Electrochem. Soc.* **133**(7), 1479 (1986).

Phillip, H. R., "Silicon Dioxide (SiO2) (Glass)", in Handbook of Optical Constants of Solids, Edward D. Palik, Ed., (Academic Press, Orlando FL, 1985).

Pierson, Arthur H., "Know Your Monochromator," *Electro-Optical Systems Design,* Feb. (1979).

Plumb, I.C., and K.R. Ryan, "Gas Phase Reactions of CF_3 and CF_2 with Atomic and Molecular Fluorine:Their Significance in Plasma Etching," *Plasma Chem. Plasma Proc.* **6**(1), 11 (1986).

Pons, M., O. Joubert, P. Paniez, and J. Pelletier, "Plasma Etching of Polymers: A Reinvestigation of Temperature Effects," *J. Appl. Phys.*, **70**(4), 2376 (1991).

Press, William H., Brian P. Flannery, Saul A. Teukolsky and William T. Vetterling, Numerical Recipies (Cambridge University Press, New York, 1988).

Raby, Bruce A., "Mass Spectrometric Study of Plasma Etching," *J. Vac. Sci. Technol.* **15**(2), 205 (1978).

Rangelow, I.W., "Computer Simulation of Line Edge Profiles Undergoing Ion Bombardment," *J. Vac. Sci. Technol. A* **1**(2), 410 (1983).

Reader, Joseph and Charles H. Corliss, "Part I. Wavelengths" in United States National Bureau of Standards: Wavelengths and Transition Probabilities for Atoms and Atomic Ions, (U.S. Government Printing Office, Washington D.C., 1980).

Richards, Albert D., "Characterization and Modeling of Polysilicon Etching in a Chlorine Plasma," Ph.D. Thesis, Massachusetts Institute of Technology, Dept. of Chemical Engineering, (1986).

Richards, Albert D., Brian E. Thompson, Kenneth D. Allen, and Herbert H. Sawin, "Atomic Chlorine Concentration Measurements in a Plasma Etcher Reactor. I. A Comparison of Infrared Absorption And optical Emission Actinometry," *J. Appl. Phys.* **62**(3), 792 (1987).

Rietman, Edward A., Robert C. Frye, Earl R. Lory, and Thomas R. Harry, "Active Neural Network Control of Wafer Attributes in a Plasma Etch Process," *J. Vac. Sci. Technol. B* **11**(4), 1314 (1993).

Roarke, Raymond J., and Warren C. Young, Formulas for Stress and Strain (McGraw-Hill, Hew York, 1975), p. 324-327,405-140.

Rogoff, G.L., J.M. Kramer, and R.B. Piejak, "A Model for the Bulk Plasma in an RF Chlorine Discharge," *IEEE Trans. Plasma Sci.* **PS-14**(2), 103 (1986).

Roland, James P., Paul J. Marcoux, Gary W. Ray, and Glenn H. Rankin, "Endpoint Detection in Plasma Etching," *J. Vac. Sci. Technol. A* **3**(3), 631 (1985).

Rossnagel, S.M., and K.L. Saenger, "Optical Emission in Magnetrons: Nonlinear Aspects," *J. Vac. Sci. Technol. A* **7**(3), 968 (1989).

Rossen, Rebecca A., and Herbert H. Sawin, "Time-of-Flight and Surface Residence Time Measurements for Ion-Enhanced Si-Cl_2 Reaction Products," *J. Vac. Sci. Technol. A* **5**(4), 1595 (1987).

Sameith, D., J.P. Mönch, H.-J. Tiller, and K. Schade, "Vibrational Analysis of the UV Emission Spectrum of Dichlorosilyl, $SiCl_2$," *Chem. Phys. Lett.* **128**(5,6) 483 (1986).

Sampson, R.K., K.A. Conrad, E.A. Irene, and H.Z. Massoud, "Simultaneous Wafer Temperature and Oxide Film Thickness Measurements in Rapid-Thermal Processing Using Ellipsometry," *J. Electrochem. Soc.* **140**(6), 1734 (1993).

Samukawa, Seiji, Yasuhiro Suzuki, and Masami Sasaki, "Extremely High-Selective Electron Cyclotron Resonance Plasma Etching for Phosphorus-Doped Polycrystalline Silion," *Appl. Phys. Lett.* **57**(4), 403 (1990).

Savas, S.E., and K.G. Donohoe, "Capacitve Probes for rf Process Plasmas," *Rev. Sci. Instrum.* **60**(11), 3391 (1989).

Sawin, Herbert H., Brian E. Thompson, and Albert D. Richards, "The Kinetics of Polysilicon Etching in Cl_2 Discharges," *Proc. Electrochem. Soc.* **85-1**, 534 (1985).

Sawin, Herbert H., and L. Rafael Reif, "Plasma Processing in Integrated Circuit Manufacturing," M.I.T. 10.616J/6.776J Course Notes, Massachusetts Institute of Technology, Cambridge, MA (1989).

Sato, Masaaki, and Yoshinobu Arita, "Etched Shape Control of Single-Crystal Silicon in Reactive Ion Etching Using Chlorine," *J. Electrochem. Soc.* **134**(11), 2856 (1987).

Sekine, M., K. Horioka, T. Arikado, Y. Muraguchi, and H. Okano, "Highly Selective Etching of Phosphorus-Doped Polycrystalline at Low Wafer Temperature Employing Magnetron Plasma," *1988 Dry Process Symposium*, 54 (1988).

Sekine, Makoto, Keiji Horioka, Tsunetoshi Arikado, and Haruo Okano, "Highly Selective Etching of Polycrystalline Silicon on Silicon Dioxide at Low Wafer Temperature, Employing Magnetron Plasma," *J. Appl. Phys.*, **73**(3), 1505(1993).

Selwyn, Gary S., "Laser Diagnostic Techniques for Reactive Ion Etching: Plasma Understanding to Process Control," *J. Vac. Sci. Technol. A* **6**(3), 2041 (1988).

Shin, Hyungcheol, Neeta Jha, Xue-Yu Qian, Graham W. Hills, and Chenming Hu, "Plasma Etching Charge-Up Damage to Thin Oxides," *Solid State Tech.* **36**(8), 29 (1993a).

Shin, Hyungcheol, Ko Noguchi, Xue-Yu Qian, Neeta Jha, Graham Hills, and Chenming Hu, "Spatial Distributions of Thin Oxide Charging in Reactive Ion Etcher," *IEEE Elec. Dev. Lett.* **14**(2), 88 (1993b).

Shone, F., K. Wu, J. Shaw, E. Hokelek, S. Mittal, and A. Haranahalli, "Gate Oxide Charging and its elimination for Metal Antenna Capacitor and Transistor in VLSI CMOS Double Layer Metal Technology," *Digest of Technical Papers, 1989 VLSI Symposium*, Kobe, Japan, 73 (1989).

Singer, Pete, "Meeting Oxide, Poly and Metal Etch Requirements," *Semicon. Intl.* **16**(4), 50 (1993).

Singh, Jyothi, R.S. Bennett, and H. Jones, "Studies of Pattern Factor Dependent Etching," SRC Topical Research Conference on Plasma Etching, J. Richard Burke, Ed., Pub. No. P890003, 273 (1989).

Singh, Vivek K., Eric S.G. Shaqfeh, and James P. McVittie, "Simulation of Profile Evolution in Silicon Reactive Ion Etching with Re-Emission and Surface Diffusion," *J. Vac. Sci. Technol. B* **10**(3), 1091 (1992).

Smith, J.M., Chemical Engineering Kinetics(McGraw-Hill, New York, 1981).

Smith, Donald L., in VLSI Electronics, Vol. 8: Plasma Processing for VLSI, Normal G. Einspruch and Dale Brown, Ed., (Academic Press, 1984), p. 260.

Smith, R., S.J. Wilde, G. Carter, I.V. Katardjiev, and M.J. Nobes, "The Simulation of Two-Dimensional Surface Erosion and Deposition Processes," *J. Vac. Sci. Technol. B* **5**(2), 579 (1987).

Stebbings, R.F., Ben R. Turner, and A.C.H. Smith, "Charge transfer in Oxygen, Nitrogen, and Nitric Oxide," *J. Chem. Phys.*, **38**(9), 2277 (1963).

Steinbrüchel, Christoph, "Universal Energy Dependence of Physical and Ion-Enhanced Chemical Etch Yields at Low Ion Energy," *Appl. Phys. Lett.* **55**(19), 1960 (1989).

Sternheim, M., W. van Gelder & A.W. Hartman, "A Laser Interferometer System to Monitor Dry Etching of Patterned Silicon," *J. Electrochem. Soc.* **130**(3), 655 (1983).

Surendra, Maheswaran, "Lagmuir Probe Theory and Its Application in Radio Frequency Excited Plasmas," Bachelor of Science Thesis, M.I.T. (1985).

Syau, Tsengyou, B. Jayant Baliga, and Raymond W. Hamaker, "Reactive Ion Etching of Silicon Trenches Using SF_6/O_2 Gas Mixtures," *J. Electrochem. Soc.* **138**(10), 3076 (1991).

Sze, S.M., VLSI Technology (McGraw-Hill, New York, 1983), p. 640.

Tachi, Shin'ichi, and Sadayuki Okudaira, "Chemical Sputtering of Silicon by F+, Cl+ and Br+ Ions:Reactive Spot Model for Reactive Ion Etching," *J. Vac. Sci. Technol. B* **4**(2), 459 (1986).

Tachi, Shinichi, Kazunori Tsujimoto, and Sadayuki Okudaira, "Low-Temperature Reactive Ion Etching and Microwave Plasma Etching of Silicon," *Appl. Phys. Lett.* **52**(8), 616 (1988).

Tachi, Shinichi, Kazunori Tsujimoto, Shin Arai, and Tokuo Kure, "Low-Temperature Dry Etching," *J. Vac. Sci. Technol. A* **9**(3), 796 (1991).

Tang, Y.S., and C.D.W. Wilkinson, "Reactive Ion Etching of Polycrystalline Silicon using $SiCl_4$," *Appl. Phys. Lett.* **58**(25), 2898 (1991).

Taur, Yuan, "0.1 μm CMOS and Beyond," MIT Microsystems Technology Laboratory VLSI Seminar, 8 Feb. 1994.

Thompson, Brian Eric, "Characterization and Modeling of Sulfur Hexafluoride Radio-Frequency Glow-Discharges for Etching Polysilicon," Ph.D. Thesis, Massachusetts Institute of Technology, Dept. of Chemical Engineering, (1986).

Thompson, Brian E., and Herbert H. Sawin, "Polysilicon Etching in SF_6 RF Discharges: Characterization and Diagnostic Measurements," *J. Electrochem. Soc.* **133**(9), 1887 (1986).

Thompson, Brian E., Herbert H. Sawin, and Donald A. Fisher, "Monte Carlo Simulation of Ion Transport Through rf Glow-Discharge Sheats," *J. Appl. Phys.* **63**(7), 2241 (1988).

Tiller, H.-J., F.-W. Breitbarth, G Pfauch, and R. Göbel, "Emission Spectroscopic Investigation of the Decomposition of CCl_4 in Plasma Etching Processes," *Beitr. Plasmaphys.* **23**(5), 519 (1983).

Tiller, H.-J., R. Görbel, and U. Führ, "Ion Extraction of Cl Containing Plasmas - Problems and Possibilities of the Investigations of Plasma Processes," *Beitr. Plasmaphys.* **24**(5), 487 (1984).

Tiller, H.-J., and S. Sameith, "Emissionspectroscopic Investigations of $SiCl_4$ and CCl_4/Si Plasmas for Etching Processes," *Contrib. Plasma. Phys.* **30**(5), 703 (1990).

TracorNorthern, "TN-6100 Series DARSS Detector Head Operator's Manual," TracorNorthern, Middleton, WI (1982).

Tsou, L.Y., "Highly Selective Reactive ion Etching of Polysilicon with Hydrogen Bromide," *J. Electrochem. Soc.* **136**(10), 3003 (1989).

Tsuzuki, T., S. Tabuchi, T. Banno, A. Kinbara, Y. Nakagawa, and T. Sukada, "Influence of Magnetic Field on the Self-Bias Potential on a Radio Frequency Powered Electrode in Radio Frequency Plasma," *J. Vac. Sci. Technol. A* **11**(3), 672 (1993).

Tyrrell, G.C., I.W. Boyd, and R.B. Jackman, "Ion Beam Assisted Etching of Silicon with Bromine," *Appl. Surf. Sci.* **43**, 439 (1989).

Uetake, Hiroaki, Takashi Matsuura, Tadahiro Ohmi, Junichi Murota, Koichi Fukuda, and Nobuo Mikoshiba, "Anisotropic Etching of n^+ Polycrystalline Silicon with High Selectivity Using a Chlorine and Nitrogen Plasma in an Ultraclean Electron Cyclotron Resonance Etcher," *Appl. Phys. Lett.* **57**(6), 596 (1990).

Ukai, K. and K. Hanazawa, "End-point Determination of Aluminum Reactive Ion Etching by Discharge Impedance Monitoring," *J. Vac. Sci. Technol.* **16**(2), 385 (1979).

Van Zwol, J., J. van Laar, A.W. Kolfschoten,and J. Dieleman, "Effects of Ar_+ Angle of Incidence on the Etching of Si With Cl_2 and Low-Energy Ar_+ Ions," *J. Vac. Sci. Technol. B* **5**(5), 1410 (1987).

Visser, R.J., "Determination of the Power and Current Densities in Argon and Oxygen Plasmas by *in situ* Temperature Measurements," *J. Vac. Sci. Technol. A* **7**(2), 189 (1989).

Vitkavage, D.J., A. Kornblit, R.A. Nicholas, and D.P. Favreau, "Dry Etching of High Aspect Ratio Contacts," *Proc. 17th Tegal Plasma Seminar*, Suppl. (1991).

Walker, Zane H., and Elmer A. Ogryzlo, "Kinetics of the Reaction of Cl Atoms with intrinsic and n+-Doped Polycrystalline Silicon," *J. Appl. Phys.* **69**(1), 548 (1991a).

Walker, Zane H., and Elmer A. Ogryzlo, "Kinetics of the Reaction of Molecular Bromine with Doped Polycrystalline Silicon," *J. Electrochem. Soc.* **138**(10), 3050 (1991b).

Watanabe, Tohru, and Yukimasa Yoshida, "Dielectric Breakdown of Gate Insulator Due to Reactive Ion Etching," *Solid State Tech.* **27**(4), 263 (1984).

Weast, Robert C., ed., CRC Handbook of Chemistry and Physics(CRC Press, Boca Raton FL, 1985).

Wilson, Ian H., Jeffrey Belson, and Orlando Auciello, "Secondary Effects in Ion Bombardment - Induced Surface Erosion", in Ion Bombardment Modification of Surfaces, O. Auciello and R. Kelly, Editors, (Elsevier, Amsterdam, 1984), p. 225.

Wolf, Stanley, and Richard N. Tauber, Silicon Processing for the VLSI Era, Volume 1:Process Technolgy(Lattice Press, Sunset Beach CA, 1986).

Wright, D.R., D.C. Hartman, U.C. Sridharan, M. Kent, T. Jasinski, and S. Kang, "Low Temperature Etch Chuck: Modeling and Experimental Results of Heat Transfer and Wafer Temperature," *J. Vac. Sci. Technol. A* **10**(4), 1065 (1992).

Yoem, Geun-Young, Yoshi Ono, and Tad Yamaguchi, "Polysilicon Etchback Plasma Process Using HBr, Cl_2, and SF_6 Gas Mixtures for Deep-Trench Isolation," J. *Electrochem. Soc.* **139**(2), 575 (1992).

Youngner, D.W., and C.M. Haynes, "Modeling Ion Beam Milling," *J. Vac. Sci. Technol.* **21**(2), 677 (1982).

Zau, Gavin C. H., Jeffrey W. Butterbaugh, Paul Rummel, and Herbert H. Sawin, "Monitoring and Control of Real Power in RF Plasma Processing," *J. Electrochem. Soc.* **138**(3), 872 (1991).

Appendix 1

AME-5000 Data Acquisition Code

This appendix contains listing of the source code used for data acquisition on the AME-5000. All of this code was written in Borland Turbo Pascal 6.0 and run on an 80286 PC. The following list outlines the program units that are listed in this section, and their function.

Gather:	Main Program	443
MyGraph:	Screen Graphics	456
DT2811TL:	A/D, D/A and Digital I/O	466
LIFER:	Laser Interferometry	470
Misc:	Assorted Utility Routines	472
Keit:	Keithley 485 Routines	488
HR640:	HR640 Monochromator Control	492
Luxt:	Luxtron 755 Temperature Monitor	507
DARSSOE:	DARSS Optical Emission Control	511
Comdel:	COMDEL RPM Control	520
VARS:	Variables, Global Definitions	525
AuxInOut:	RS232 Code	529

The listings for each of these units follows.

```
program Gather;

{
        Automated Data Collection for the AME5000

    Timothy J. Dalton
    MIT, 66-225
    (617) 258-8840

    April - May, 1991

}

Uses Graph, Dos, Crt, MyGraph, DT2811TL, LIFER, Misc,
   Keit, HR640, Luxt, DARSSOE, Comdel, VARS;

VAR
  Collect          : Boolean;


procedure Init;

Var
  Pos              : Integer;
  Good             : Boolean;
  X,Y              : byte;
  Ch               : String;
  Cha              : Char;

begin
  TextBackGround(LightGray);
  ClrScr;
  TextColor(Red+Blink);
  CenterText(1,'Initialization Screen');
  Writeln; writeln;
  Pos := 14;
  Dout(0);

  if (status.LUXTRON) then begin
    InitCommun;
    Writeln (IeeeOut, 'RESET');
    Close (IeeeOut);
    Close (IeeeIn);
    InitCommun;
    CheckLuxtronStatus(RSV,ERR,RDY,MC,Mode);
    InitLuxtron;
    CheckLuxtronStatus(RSV,ERR,RDY,MC,Mode);
  end;

  if (status.THR640) then begin
    Status.StartingWaveLength := -1.0;
    { Assume Nothing About the Grating and Mirror! }
    Status.THR640MirrorPos := ' ';
    Status.THR640Grating   := ' ';
```

```
TextColor(Red);
if not(status.LUXTRON) then  begin
  InitCommun;          { Open IEEE-488 }

  if (status.thr640pmt.detector = 0) then
     CloseKeithley;

  InitCommun;
end;

if (status.thr640pmt.detector = 0) then
   InitKeithley;         { Get the Keithley Set }

InitCom1Port;           { Open Lines to SpectraLink }
CenterText(Pos, 'T H R - 6 4 0');
Pos := Pos + 2;
Good := False;

{ Only Ask to Position Mirror if it has not been done! }

if ( ((Status.PMT) and (Status.THR640MirrorPos <> 'L')) or
    ((Status.LDA) and (Status.THR640MirrorPos <> 'A')) ) then begin
  Repeat
    TextColor(Blue);
    GotoXY(1,Pos); Write('Position Mirror ? (Y/N) : ');
    TextColor(Black);
    WriteBlanks(3,x,y);
    EnterText(Ch, X, Y, 1);
    Cha := UpCase(Ch[1]);
    If (Cha = 'Y') or (Cha = 'N') then
      begin
        Good := True;
        GotoXY(X,Y);
        Write(Cha);
      end
    Else
      Good := False;
  Until Good;
  If (Cha = 'Y') then begin
    if (Status.PMT) then begin
      Pos := Pos + 1;
      GotoXY(1,Pos); Write('Moving Mirror to Lateral Exit for PMT');
      Status.THR640MirrorPos := 'L';
      MCCMotorMove(ExitMirror, LateralExit); end
    else begin
      Pos := Pos + 1;
      GotoXY(1,Pos); Write('Moving Mirror to Axial Exit for LDA');
      Status.THR640MirrorPos := 'A';
      MCCMotorMove(ExitMirror, AxialExit);
    end;
    end {if Cha}
  else begin
    { If they don't wnt to move it, assume its correct! }
    If (Status.PMT) then
```

```
                Status.THR640MirrorPos := 'L'
          else
             Status.THR640MirrorPos := 'A',
        end; {if Cha}
     End; {If Status.PMT.....}

     Pos := Pos + 1;
     Good := False;

     { Only Ask to Position Mirror if it has not been done! }
     if (Status.Grating <> Status.THR640Grating) then begin
       Repeat
         TextColor(Blue);
         GotoXY(1,Pos); Write('Position Grating on Turret ? (Y/N) : ');
         TextColor(Black);
         WriteBlanks(3,x,y);
         EnterText(Ch, X, Y, 1);
         Cha := UpCase(Ch[1]);
         If (Cha = 'Y') or (Cha = 'N') then
           begin
             Good := True;
             GotoXY(X,Y);
             Write(Cha);
           end
         Else
           Good := False;
       Until Good;
       If (Cha = 'Y') then begin
         if (Status.Grating = 'A') then begin
           Pos := Pos + 1;
           GotoXY(1,Pos); Write('Changing To Grating A');
           Status.THR640Grating := 'A';
           MCCMotorMove(GratingTurret, GratingA); end
         else begin
           Pos := Pos + 1;
           GotoXY(1,Pos); Write('Changing To Grating B');
           Status.THR640Grating := 'B';
           MCCMotorMove(GratingTurret, GratingB);
         end;
         end {if Cha}
       else
         { If they don't want to move the grating, Assume its correct! }
         Status.THR640Grating := Status.Grating;
    end; {If status.grating.....}

  end; {If Status.THR640}
  Pos := Pos + 2; GotoXY(1,Pos); TextColor(Black);

  If (Status.Comdel) then begin
    InitCom2Port;
    ClearCom2;
    SetUpComdel;
  end;
```

```
  if (Status.DARSS) then begin
    DARSSParam.Started := False;
    DARSSParam.PointsToRead := 128;
    InitCalib;
    CalcWave;
  end;
  ClrScr;
  ShowLifer := False;
end;


procedure StartCollection;
var good:boolean;  i,j : Integer; Ch : Char;
   Chs : String;
   X,Y : Byte;

begin
  if (not(Collect)) then begin
    if (Status.Files = Closed) then begin
      FileNameGen(Good);
      if (Good) then begin
        OpenOutPut;
        CloseGraph;
        TextBackground(LightGray);
        ClrScr;
        TextColor(Blue);
        Writeln(InfoFile,'***** Begin Pre-Run Comments *****');
        GetInfo;
        Writeln(InfoFile,'***** End Pre-Run Comments *****');
        ClrScr;

        Repeat
          GotoXY(1,3);
          TextColor(Blue);
          write('Synchronize to Magnetic Field ? (Y/N) : ');
          TextColor(Black);
              WriteBlanks(3,x,y);
              EnterText(Chs, X, Y, 1);
              Chs[1] := UpCase(Chs[1]);
        Until ((Chs[1] = 'Y') or (Chs[1] = 'N'));
        Writeln; Writeln;
        ClrScr;
            if (chs[1] = 'Y') then begin
              MagSync := 1;

          repeat
            writeln('Enter the # of 1/2 periods to Sync to: 1 or 2 : ');
            readln(magmodnum);
          until ( (magmodnum = 1) or (magmodnum = 2) );
          ClrScr;

          TextColor(Red);
          writeln('Magnetic Field Synchronization Activated');
          writeln(magmodnum,' half-periods will be synced to.');
```

```
  DtoA(0,5.0);
      end
    else
      MagSync := 0;


Repeat
  GotoXY(1,5);
  TextColor(Blue);
  write('SET PMT HV Early ? (Y/N) : ');
  TextColor(Black);
      WriteBlanks(3,x,y);
      EnterText(Chs, X, Y, 1);
      Chs[1] := UpCase(Chs[1]);
Until ((Chs[1] = 'Y') or (Chs[1] = 'N'));
Writeln; Writeln;
    if (chs[1] = 'Y') then begin
  InitCom1Port;     { Open Lines to SpectraLink }
      GetVolt;
end;

if (status.thr640) then begin
       ClrScr;
   repeat
     GotoXY(1,5);
     TextColor(Blue);
     writeln('Use Which Detector ?     ');
     writeln;
     writeln('0 - Keithley Picoameter on IEEE488');
     writeln('1 - Fast Current to Voltage Converter on ADC 1');
     writeln;
     TextColor(Black); GotoXY(22,5);
         WriteBlanks(3,x,y);
         EnterText(Chs, X, Y, 1);
       Until ((Chs[1] = '0') or (Chs[1] = '1'));
   GotoXY(1,9);
   if (Chs[1] = '0') then begin
      Status.THR640PMT.Detector := 0;
      writeln('KEITHLEY CHOSEN');
    end
   else begin
      Status.THR640PMT.Detector := 1;
      writeln('FAST I -> V CHOSEN');
      repeat
        GotoXY(1,10);
        TextColor(Blue);
        writeln('Enter Gain Setting (1,2,4,8):       ');
        writeln;
        TextColor(Black); GotoXY(34,10);
            WriteBlanks(3,x,y);
            EnterText(Chs, X, Y, 1);
          Until ((Chs[1] = '1') or (Chs[1] = '2') or
                  (Chs[1] = '4') or (Chs[1] = '8'));
```

```
    case Chs[1] of
          '1' : Status.Thr640PMT.Gain := Gain_1;
          '2' : Status.Thr640PMT.Gain := Gain_2;
                  '4' : Status.Thr640PMT.Gain := Gain_4;
          '8' : Status.Thr640PMT.Gain := Gain_8;
    end; {case}

  end;

  writeln; writeln('Enter K427 Gain Setting, x, in 10^x');
  readln(status.THR640PMT.Gain2);
  writeln; writeln('Enter K427 Rise Time in mSec      ');
  readln(status.THR640PMT.RiseTime);

  writeln; writeln('Press Any Key to Continue');
   end; {if status.thr640}
Repeat Until KeyPressed;


ClrScr;
TextColor(Red);
if (status.LUXTRON) then LuxStart := False;

If (Status.THR640) then begin
      If (Status.PMT) then begin


        if (Status.THR640PMT.WaveScan) then begin
      { Have Any Parameters Been Entered Yet ? }
      If (Status.StartingWaveLength <> -1.0) then begin
        Repeat
             GotoXY(1,3);
          write('Change WaveScan Parameters ? (Y/N) : ');
          TextColor(Black);
                  WriteBlanks(3,x,y);
                  EnterText(Chs, X, Y, 1);
                  Chs[1] := UpCase(Chs[1]);
        Until ((Chs[1] = 'Y') or (Chs[1] = 'N'));
      End;
      if ((Chs[1] = 'Y') or
                (Status.StartingWaveLength = -1.0)) then Begin
        Writeln; Writeln;
                GetWaveScanParam;
        GetKeitAvg;
        GetVolt;
        GetSlit;
        MoveToStartingPosition;
        end
      else begin
        SetVoltage(Status.PMTVoltage, Terminate);
        MoveToStartingPosition;
      end;

    end; {If Status.THR640PMT.WaveScan}
```

```
          if (Status.THR640PMT.OptEmission) then begin

      Repeat
          GotoXY(1,5);
        write('Move Monochromator ? (Y/N) : ');
        TextColor(Black);
                WriteBlanks(3,x,y);
                EnterText(Chs, X, Y, 1);
                Chs[1] := UpCase(Chs[1]);
      Until ((Chs[1] = 'Y') or (Chs[1] = 'N'));
      Writeln; Writeln;

      if (Chs[1] = 'Y') then Begin
                GetStartingWave;
        GetVolt;
        GetSlit;
        MoveToStartingPosition;
        end
      else begin
        If (Status.StartingWaveLength = -1.0) then begin
          ReadMeter;
          Status.StartingWaveLength := WaveLength;
        end;
        GetVolt;
        GetSlit;
      end;
    End;
      End;
    End;


if (Status.DARSS) then begin
  CenterText(2,'DARSS Background Scan');
  CenterText(5,'Press Return For BackGround Scan!');
  Repeat Until KeyPressed;
  CenterText(8,'S C A N N I N G');
  ch := ReadKey;
  { Get A Background }
  StartDarss(DARSSParam.Exposure);
  Repeat Until IsDARSSReady;
  ReadDataFromDarss(1024, DARSSParam.BData);
  StartDarss(DARSSParam.Exposure);
  Repeat Until IsDARSSReady;
  ReadDataFromDarss(1024, DARSSParam.BData);
  { Keep the Last One }
  StartDarss(DARSSParam.Exposure);
  Repeat Until IsDARSSReady;
  ReadDataFromDarss(1024, DARSSParam.BData);
  Writeln(OptFile,'*****');
  Writeln(OptFile,'INITIAL BACKGROUND SCAN');
  i := 1;
```

```
      While (i < 1022) do begin
            Write(OptFile, i:4,' ');
        For j := 0 to 7 do
          write(OptFile,DarssParam.bdata[i+j]:5,' ');
        i := i + 8;
        Writeln(OptFile);
      end;
      ClrScr;
    end; { if DARSS }

    end
  else begin
    ClrScr;
    Writeln('Problem Generating File Name');
    exit;
  end;
end; { if Status.Files }
Collect := True;
GrInit; InitMenu; DrawMenu;
WriteInfo;
Prompt;
ShowFileName;
REPEAT Until KeyPressed;
ClearMenuArea;
PromptGathering;
ShowFileName;
if (Status.LIFER) then
  Acq_Lifer;
if ((status.LUXTRON) and not(status.lifer)) then Acq_Temp;

if (Status.THR640) then begin
  Status.THR640PMT.CTR := 0;
  if (Status.THR640PMT.WaveScan) then begin

    if (Status.THR640PMT.Detector = 0) then
       Status.THR640PMT.MAX := 50.0
    else
       Status.THR640PMT.MAX := 0.5;

    SetUp1( Status.StartingWaveLength, Status.EndingWaveLength,
         (Status.EndingWaveLength-Status.StartingWaveLength)/10.0,
         0.0, Status.THR640PMT.MAX, Status.THR640PMT.MAX/10.0, 0);
    end
  else begin
    Status.THR640PMT.XMin := 0.0;
    Status.THR640PMT.XMax := 120.0;
    if (Status.THR640PMT.Detector = 0) then
       Status.THR640PMT.MAX := 50.0
    else
       Status.THR640PMT.MAX := 0.5;
    SetUp1( Status.THR640PMT.XMin, Status.THR640PMT.XMax, 20.0,
         0.0, Status.THR640PMT.MAX, Status.THR640PMT.MAX/10.0, 1);
  end;
end; {If Status.THR640}
```

```
    Delay(10);
    Dout(1);
    MyGetTime(Time0);
    Delay(5);
    if (MagSync = 1) then begin
      check_field;
          num_rise := 0; prev_rise := 0;
    end;

  end; { if not(Collect) }
end; { Start Collection }


Procedure EndCollection;
var I,j : integer; S : String;
   Time, Inten, TMax : Real;

BEGIN
  Dout(0);
  if (Status.Files = Open) then begin
    {
     If we are using the DARSS, then check to see if the data
         has finished being read in.  If it has not, call the
     DARSS control routine until it is finished.

     This is to remove a bug with partially read in data if
     the exit key is hit during a run at the wrong time.
    }
    if (Status.DARSS) then begin
      while (not(DARSSParam.DataRead)) do
        ControlDARSS;
    end;
    { Now Continue with the End Collection Routine! }
    CloseGraph;
    TextBackground(LightGray);
    ClrScr;
    TextColor(Blue);
    Writeln(InfoFile,'***** END OF RUN COMMENTS *****');
    GetInfo; ClrScr; TextColor(Red);

    if (Status.DARSS) then begin
      CenterText(2,'DARSS Background Scan');
      CenterText(5,'Press Return For BackGround Scan!');
      Repeat Until KeyPressed;
      CenterText(8,'S C A N N I N G');
      ch := ReadKey;
      StartDarss(DARSSParam.Exposure);
      Repeat Until IsDARSSReady;
      ReadDataFromDarss(1024, DARSSParam.BData);
      Writeln(OptFile,'*****');
      Writeln(OptFile,'FINAL BACKGROUND SCAN');
      i := 1;
      While (i < 1022) do begin
            Write(OptFile, i:4,' ');
```

```
    For j := 0 to 7 do
      write(OptFile,DarssParam.bdata[i+j]:5,' ');
    i := i + 8;
    Writeln(OptFile);
  end;
  ClrScr;
  DARSSParam.Started := False;
end; {If Status.DARSS}

if (Status.Files = Open) Then
  CloseOutPut;

Collect := False;
ClrScr;
GrInit; InitMenu; DrawMenu;

if (Status.LIFER) then begin
  View2;
  ClearViewPort;
  SetUp2(0.0, LiferTMax, LiferTMax/10.0, GR2YMin, GR2YMax, GR2dY);
  Assign(LiferFile,Status.Path+LiferPath+Status.FileBase+'.LIN');
  Reset(LiferFile);
  Readln(LiferFile, Time, Inten);
  While ( (Time < LiferTMax) and
        (IOResult = 0) and not(eof(Liferfile)) ) do begin
    View2;
    PutPixel( View2X(Time), View2Y(Inten), green);
    Readln(LiferFile, Time, Inten);
  end;
  Close(LiferFile);
end;

  if (Status.THR640) then begin
  HVOff(Terminate);
  StopGrating;

  If (Status.THR640PMT.WaveScan) Then Begin
    View1;
    ClearViewPort;
    SetUp1( Status.StartingWaveLength, Status.EndingWaveLength,
        (Status.EndingWaveLength-Status.StartingWaveLength)/10.0,
        0.0, Status.THR640PMT.MAX, Status.THR640PMT.MAX/10.0, 0);
    For I := 1 to Status.THR640PMT.Ctr do
      PutPixel( View1X(DarssParam.Wave[i]),
                View1Y(DarssParam.SData[i]), White);
    end
  else begin
    View1;
    ClearViewPort;
    Status.THR640PMT.XMax := DARSSParam.Wave[Status.THR640PMT.CTR];
    SetUp1( 0.0, Status.THR640PMT.XMax, Status.THR640PMT.XMax/10.0,
        0.0, Status.THR640PMT.MAX, Status.THR640PMT.MAX/10.0, 1);
    Assign(OptFile,Status.Path+OptPath+Status.FileBase+'.OPT');
    Reset(OptFile);
```

```
        Readln(OptFile, Time, Inten);
        While ( (Time < Status.THR640PMT.XMax) and
             (IOResult = 0) and not(eof(Optfile)) ) do begin
          View1;
          PutPixel( View1X(Time), View1Y(Inten), White);
          Readln(OptFile, Time, Inten);
        end;
        Close(OptFile);
      end;
    end;
  end; {if Status.Files}
end; { Proc }


procedure FinishUp;
begin
  if (Status.Files = Open) then
    EndCollection;
  If (Status.Comdel) then CloseCom2Port;
  CloseGraph;
  if (status.LUXTRON) then CloseLuxtron;
  If (Status.THR640) then begin
    HVOff(Terminate);
    if (status.thr640pmt.detector = 0) then
      CloseKeithley;        { Keithly Reset }
    CloseCom1Port;          { Close SpectraLink}
  end;
  CloseComm;    { Turn Off IEEE488 Link }
end;


procedure ControlAll;
var done : boolean;  ch : char; S : String;
   ready:integer; i : integer;
begin
  Dout(0);
  done := false;
  Collect := False;
  while not(done) do begin
    if KeyPressed then begin
      ch := ReadKey;
      if ch = #0 then begin
        ch := ReadKey;
        case ch of
          #59  : StartCollection; { F1 : Begin Data Collection }

          #60  : EndCollection; { F2 : End Data Collection }

          #61  : Begin { F3 : Show Interferometry }
                      ShowLifer := True;
                MyGetTime(Time0);
                Acq_Lifer;
              end;
```

```
    #62  : begin { F4 : Stop Showing Interferometry }
              ShowLifer := False;
           View2;
           ClearViewPort;
         end;

    #63  : begin { F5 Change Params }
                  if (not(Collect)) then begin
            StartUp; Init;
                        GrInit; InitMenu; DrawMenu;
          end;
         end;

    #67  : begin { F9 Test Stuff! }
           View1;
           ClearViewPort;
           If (Collect) then
             S := 'F9 TEST   Collect = TRUE'
           else
             S := 'F9 TEST   Collect = FALSE';
           OutTextXY(5,5,S);
         end;

    #68  : begin { F10 : Exit! }
                  { Only Exit if No Data Files Open }
           { Must Stop Data Gathering Before Exiting }
                  if (Status.Files = Closed ) then
                    Done := True;
         end;
   end; {Case}

  end; {if ch = #0}
end; {If KeyPressed}

With STATUS do begin
  if ( (COLLECT) and (not(done)) ) then begin

       if (MagSync = 1) then begin
     Check_Field;
      if ( ( (num_rise mod MagModNum) = 0) and (num_rise <> prev_rise) ) then
        ready := 1
     else
        ready := 0;
     end
   else
     ready := 1;

       if (READY = 1) then begin
     if (LIFER) then GetLiferPoint;
     if (DARSS) then ControlDARSS;
     if (COMDEL) then GetComdelPoint;
     if (THR640) then begin
           ControlTHR640(Terminate);
       if (Terminate) then EndCollection;
```

```
          Terminate := False;
        end;
        if (LUXTRON) then ControlLuxtron;
      end;

      { i := 0;
       repeat
       i := i + 1;
       until (i = 30000); }

    end; {if COLLECT}
  end; {With}

  if ( (ShowLifer) and not(COLLECT) ) then GetLiferPoint;

 end; {While not(done)}
end; {proc}


begin
 StartUp;
 Init;
 GrInit;
 InitMenu;
 DrawMenu;
 ControlAll;
 FinishUp;
end.
```

```
unit MyGraph;
{
        The Constants in This Section are Defined for a ***VGA***
     Screen.  The output will be ***INCORRECT*** on other screen types
}

interface

Uses Graph, Dos, Crt, DT2811TL, Misc, VARS;

const
  { Define Graph 1 & 2  X&Y Start and End Points }
  Gr1XS  : integer = 0;
  Gr1YS  : integer = 15;
  Gr1XE  : integer = 500;
  Gr1YE  : integer = 247;
  Gr2XS  : integer = 0;
  Gr2YS  : Integer = 247;
  Gr2XE  : Integer = 639;
  Gr2YE  : Integer = 479;
  MenuXS : Integer = 500;
  MenuYS : Integer = 15;
  MenuXE : Integer = 639;
  MenuYE : Integer = 247;

  Grid   : Boolean = True;

var graphdriver, graphmode: integer;
   xm,ym : integer;
   v2xs, v2ys, v2xe, v2ye : integer;
   v1xs, v1ys, v1xe, v1ye : integer;
   Size2X, Size2Y : integer;
   Size1X, Size1Y : integer;
   Min2X, Max2X, dx2, Min2Y, Max2Y, dy2: real;
   XView2Min, XView2Max, YView2Min, YView2Max : integer;
   Min1X, Max1X, dx1, Min1Y, Max1Y, dy1: real;
   XView1Min, XView1Max, YView1Min, YView1Max : integer;
   MXS, MYS, MXE, MYE, SizeMX, SizeMY : integer;

   { List All The Procedures }
   procedure CenterTextX(xmax, y:integer; s:string);
   procedure CenterTextY(x, ymax:integer; s:string);
   function  View1Y(Y:real):integer;
   function  View1X(X:real):integer;
   function  View2Y(Y:real):integer;
   function  View2X(X:real):integer;
   procedure GrInit;
   procedure Gr1View(MinX, MaxX, dX, MinY, MaxY, dY: Real);
   procedure Gr2View(MinX, MaxX, dX, MinY, MaxY, dY: Real);
   Procedure View2;
   Procedure View1;
   procedure SetUp2(MinX, MaxX, dX, MinY, MaxY, dY: Real);
   procedure SetUp1(MinX, MaxX, dX, MinY, MaxY, dY: Real; TitleType:Integer);
   procedure InitMenu;
```

```
   procedure ClearMenuArea;
   procedure DrawMenu;
   Procedure PROMPT;
   procedure ShowFileName;
   procedure PromptGathering;
   procedure EnterString(var s:string; x,y:integer; max : integer);

 Implementation

procedure CenterTextX(xmax, y:integer; s:string);
{ Center a String in the X Direction }
begin
 OutTextXY(((xmax - TextWidth(s)) div 2),y,s);
end;

procedure CenterTextY(x, ymax:integer; s:string);
{ Center a String in the Y Direction }
begin
 OutTextXY(x,((ymax - TextWidth(s)) div 2),s);
end;

{ Convert From Window Coordinates(real) to Viewport Coordinates(Int) }
function View2Y(Y:real):integer;
var m : real;
begin
 m := (YView2Max - YView2Min) / (Max2Y - Min2Y);
 View2Y := YView2Min + round(m * (Y-Min2Y));
end;

function View2X(X:real):integer;
var m : real;
begin
 m := (XView2Max - XView2Min) / (Max2X - Min2X);
 View2X := XView2Min + round(m * (X-Min2X));
end;


{ Convert From Window Coordinates(real) to Viewport Coordinates(Int) }
function View1Y(Y:real):integer;
var m : real;
begin
 m := (YView1Max - YView1Min) / (Max1Y - Min1Y);
 View1Y := YView1Min + round(m * (Y-Min1Y));
end;

function View1X(X:real):integer;
var m : real;
begin
 m := (XView1Max - XView1Min) / (Max1X - Min1X);
 View1X := XView1Min + round(m * (X-Min1X));
end;

procedure GrInit;
{Initialize Graphics}
```

```
begin
 graphdriver := detect;
 InitGraph(graphdriver, graphmode, '');
 if (graphresult <> grok) then
   halt(2);
 ClearDevice;
 xm := GetMaxX;
 ym := GetMaxY;
 SetColor(White);
 Rectangle(0,0,Xm,Ym); { Outline Entire Screen }
 MoveTo(0,15); LineTo(xm,15); { Draw a Line to Separate Titles }
 SetColor(LightRed);
 Rectangle(Gr1XS, Gr1YS, Gr1XE, Gr1YE); { 1st Graph Boundry }
 SetColor(LightGreen);
 Rectangle(Gr2XS, Gr2YS, Gr2XE, Gr2YE); { 2nd Graph Boundry }
 SetColor(LightCyan);
 if (Get_LIFER) then
 CenterTextX(xm,4,'AME-5000 Data Acquisition System: Timothy J. Dalton (c) 1991');
end;

procedure View1;
begin
  SetViewport(v1xs, v1ys, v1xe, v1ye, ClipOn);
end;

procedure View2;
begin
  SetViewport(v2xs, v2ys, v2xe, v2ye, ClipOn);
end;


procedure Gr2View(MinX, MaxX, dX, MinY, MaxY, dY: Real);
begin
  { Calc. Allowed Viewport For Graph 2 }
  v2xs := Gr2XS + 1;
  v2ys := Gr2YS + 1;
  v2xe := Gr2XE - 1;
  v2ye := Gr2YE - 1;
  { Calc. Size of Window in Pixels }
  Size2X := v2xe - v2xs;
  Size2Y := v2ye - v2ys;
  { Store Calling Parameters }
  Min2X := MinX; Max2X := MaxX; dx2 := dx;
  Min2Y := MinY; Max2Y := Maxy; dy2 := dy;
  SetViewport(v2xs, v2ys, v2xe, v2ye, ClipOn);
  ClearViewPort;
  { Calc. Graph Drawing Limits Leaving Rooms For Axes & Labels }
  XView2Min := 0 + (Size2X div 10);
  YView2Max := 0 + (Size2Y div 10);
  XView2Max := Size2X - ((Size2X * 4) div 100);
  YView2Min := Size2Y - (Size2Y div 8);
  { Rectangle(XView2Min, YView2Min, XView2Max, YView2Max);}
end;
```

```
procedure Gr1View(MinX, MaxX, dX, MinY, MaxY, dY: Real);
begin
  { Calc. Allowed Viewport For Graph 1 }
  v1xs := Gr1XS + 1;
  v1ys := Gr1YS + 1;
  v1xe := Gr1XE - 1;
  v1ye := Gr1YE - 1;
  { Calc. Size of Window in Pixels }
  Size1X := v1xe - v1xs;
  Size1Y := v1ye - v1ys;
  { Store Calling Parameters }
  Min1X := MinX; Max1X := MaxX; dx1 := dx;
  Min1Y := MinY; Max1Y := Maxy; dy1 := dy;
  SetViewport(v1xs, v1ys, v1xe, v1ye, ClipOn);
  ClearViewPort;
  { Calc. Graph Drawing Limits Leaving Rooms For Axes & Labels }
  XView1Min := 0 + (Size1X div 10);
  YView1Max := 0 + (Size1Y div 10);
  XView1Max := Size1X - ((Size1X * 4) div 100);
  YView1Min := Size1Y - (Size1Y div 8);
  { Rectangle(XView2Min, YView2Min, XView2Max, YView2Max);}
end;


{ Given Real World Coordinate Limits, Draw Graph, including axes,
  labels for numbers and titles }
procedure SetUp2(MinX, MaxX, dX, MinY, MaxY, dY: Real);
var a,b,c,d,XStart,YStart : integer; s,s1 : string; e: real;

function XAxisVisible:boolean;
begin
 if (MinY <= 0) and (MaxY >= 0) then
   XAxisVisible := True
 else
   XAxisVisible := False;
end;

function YAxisVisible:boolean;
begin
 if (MinX <= 0) and (MaxX >= 0) then
   YAxisVisible := True
 else
   YAxisVisible := False;
end;

begin
  Gr2View(MinX, MaxX, dX, MinY, MaxY, dY);

  if XAxisVisible then
    YStart := View2Y(0.0)
  else
    YStart := YView2Min;
  a := View2X(MinX);
  b := View2X(MaxX);
```

```
SetColor(White);
c := YStart;
MoveTo(a,c);
LineTo(b,c);
e := MinX;
while e <= MaxX do begin
  b := View2X(e);
  MoveTo(b,c);
  d := Size2Y div 50;
  SetColor(White);
  LineTo(b,(c+d));
  str(e:5:1,s);
  SetColor(LightCyan);
  OutTextXY( b-TextWidth(s) div 2,c+2*d,s);
  if grid then begin
    SetColor(LightGray);
    MoveTo(b,YView2Min);
    LineTo(b,YView2Max);
  end;
  e := e + dx;
end;
SetColor(Brown);
CenterTextX(size2x,Size2Y-10,'Time (Seconds)');

if YAxisVisible then
  XStart := View2X(0.0)
else
  XStart := XView2Min;
a := View2Y(MinY);
b := View2Y(MaxY);
SetColor(White);
c := XStart;
MoveTo(c,a);
LineTo(c,b);
e := MinY;
while e <= MaxY do begin
  b := View2Y(e);
  MoveTo(c,b);
  d := Size2X div 100;
  SetColor(White);
  LineTo((c-d),b);
  str(e:5:1,s);
  SetColor(LightCyan);
  OutTextXY( c-2*d-TextWidth(s),b-TextHeight('A') div 2,s);
  if grid then begin
    SetColor(LightGray);
    MoveTo(XView2Min,b);
    LineTo(XView2Max,b);
  end;

  e := e + dy;
end;
SetTextStyle(DefaultFont,VertDir,1);
SetColor(Brown);
```

```
  CenterTextY(14,Size2Y,'Ch0 Input Signal/2 (Volts)');
  SetTextStyle(DefaultFont,HorizDir,1);

  if (Get_LIFER) then begin
    SetColor(LightCyan);
    s := 'Laser Interferometry Signal' + ' : ' + OutFileString;
    CenterTextX(Size2x,4,s);
  end;

end;

{ Given Real World Coordinate Limits, Draw Graph, including axes,
  labels for numbers and titles }
procedure SetUp1(MinX, MaxX, dX, MinY, MaxY, dY: Real; TitleType : Integer);
var a,b,c,d,XStart,YStart : integer; s,s1 : string; e: real;

function XAxisVisible:boolean;
begin
 if (MinY <= 0) and (MaxY >= 0) then
   XAxisVisible := True
 else
   XAxisVisible := False;
end;

function YAxisVisible:boolean;
begin
 if (MinX <= 0) and (MaxX >= 0) then
   YAxisVisible := True
 else
   YAxisVisible := False;
end;

begin
  SetTextStyle(SmallFont, HorizDir, 4);
  Gr1View(MinX, MaxX, dX, MinY, MaxY, dY);

  if XAxisVisible then
    YStart := View1Y(0.0)
  else
    YStart := YView1Min;
  a := View1X(MinX);
  b := View1X(MaxX);
  SetColor(White);
  c := YStart;
  MoveTo(a,c);
  LineTo(b,c);
  e := MinX;
  while e <= MaxX do begin
    b := View1X(e);
    MoveTo(b,c);
    d := Size1Y div 50;
    SetColor(White);
    LineTo(b,(c+d));
    str(e:5:1,s);
```

```
    SetColor(LightCyan);
    OutTextXY( b-TextWidth(s) div 2,c+2*d,s);
    if grid then begin
      SetColor(LightGray);
      MoveTo(b,YView1Min);
      LineTo(b,YView1Max);
    end;
    e := e + dx;
  end;
  SetColor(Brown);
  if (TitleType = 0) then
    CenterTextX(size1x,Size1Y-15,'WaveLength(A)')
  else
    CenterTextX(size1x,Size1Y-15,'Time (Seconds)');

  if YAxisVisible then
    XStart := View1X(0.0)
  else
    XStart := XView1Min;
  a := View1Y(MinY);
  b := View1Y(MaxY);
  SetColor(White);
  c := XStart;
  MoveTo(c,a);
  LineTo(c,b);
  e := MinY;
  while e <= MaxY do begin
    b := View1Y(e);
    MoveTo(c,b);
    d := Size1X div 100;
    SetColor(White);
    LineTo((c-d),b);
    str(e:5:1,s);
    SetColor(LightCyan);
    OutTextXY( c-2*d-TextWidth(s),b-TextHeight('A') div 2,s);
    if grid then begin
      SetColor(LightGray);
      MoveTo(XView1Min,b);
      LineTo(XView1Max,b);
    end;

    e := e + dy;
  end;
  SetTextStyle(DefaultFont,VertDir,1);
  SetColor(Brown);
  CenterTextY(14,Size1Y,'Intensity');
  SetTextStyle(DefaultFont,HorizDir,1);

end;

procedure InitMenu;
begin
  MXS := MenuXS + 1;
  MYS := MenuYS + 1;
```

```
  MXE := MenuXE - 1;
  MYE := MenuYE - 1;
  SizeMX := MXE - MXS;
  SizeMY := MYE - MYS;
end;

procedure ClearMenuArea;
begin
 SetViewport(MXS, MYS, MXE, MYE, ClipOn);
 ClearViewPort;
end;


procedure Prompt;
var pos, delta,delta2 : integer;
begin
  delta2 := TextHeight('A');
  delta := delta2 + TextHeight('A') div 2;
  ClearMenuArea;
  SetColor(Yellow);
  OutTextXY(5,5*Delta,'Press ENTER');
  OutTextXY(5,6*Delta,'To Begin');
end;


procedure ShowFileName;
var pos, delta,delta2 : integer;
begin
  delta2 := TextHeight('A');
  delta := delta2 + TextHeight('A') div 2;
  SetColor(WHITE);
  OutTextXY(5,12*Delta,'Data File:');
  OutTextXY(5,13*Delta,Status.FileBase);
end;



procedure PromptGathering;
var pos, delta,delta2 : integer;
begin
  delta2 := TextHeight('A');
  delta := delta2 + TextHeight('A') div 2;
  ClearMenuArea;
  SetColor(Yellow);
  OutTextXY(5,5*Delta,'GATHERING');
  OutTextXY(5,6*Delta,'DATA');
  SetColor(White);
  OutTextXY(5,2*Delta,'F2 to End');
end;


procedure DrawMenu;
var pos, delta,delta2 : integer;
begin
```

```
  delta2 := TextHeight('A');
  delta := delta2 + TextHeight('A') div 2;
  pos := 4;
  ClearMenuArea;
  SetColor(Green);
  CenterTextX(SizeMX, pos, '* MENU *');
  SetColor(Yellow);
  inc(pos,delta);
  OutTextXY(5,pos,'F1 : Acquire');
  inc(pos,delta2);
  OutTextXY(5+TextWidth('F1 : '),pos,'Data');
  inc(pos,delta);
  OutTextXY(5,pos,'F2 : Stop Data');
  inc(pos,delta2);
  OutTextXY(5+TextWidth('F2 : '),pos,'Acquire');
  inc(pos,Delta);
  OutTextXY(5,pos,'F3 : View Lifer');
  inc(pos,delta);
  OutTextXY(5,pos,'F4 : Stop Lifer');
  inc(pos,delta);
  OutTextXY(5,pos,'F5 : Change Param');
  inc(pos,delta);
  OutTextXY(5,pos,'F10: Exit');
  pos := MYE - 2*delta;
(* SetColor(Green);
  OutTextXY(5,pos,'Reserved');
  dec(pos,delta2);
  OutTextXY(5,pos,'Reserved');
  dec(pos,delta);
  OutTextXY(5,pos,'Reserved');
  SetColor(White);
  MoveTo(1,pos-2);
  LineTo(SizeMX-1,pos-2);
  Inc(pos,delta2);
  SetColor(Green);
  MoveTo(1,pos+1);
  LineTo(SizeMX-1,pos+1); *)


end;

procedure EnterString(var s:string; x,y:integer; max : integer);
var loc:byte; done : boolean; ch : char;
begin
  done := false;
  loc := 0;
  while not(done) do begin
    if Keypressed then begin
      ch := ReadKey;
      if ( ((ch >= 'a') and (ch <= 'z')) or ((ch >= 'A') and (ch <= 'Z'))
           or ((ch >= '0') and (ch <= '9')) or (ch = '.') ) then begin
        if (loc < max) then begin
              inc(loc);
          s[loc] := ch;
```

```
            OutTextXY(x,y,ch);
          x := x + TextWidth(ch)
        end {if loc}
        end {if ch....}
      else
      if (ch = ^H) then
            begin
          if (loc <> 0) then begin
            x := x - TextWidth(S[loc]);
            SetColor(0);
            OutTextXY(x,y,S[loc]);
            SetColor(Green);
            dec(loc);
          end {if loc<>0}
        end {if ch =^H}
      else
        if (ch = ^M) then begin
          done := true;
          s[0] := chr(loc);
        end; {if ch = ^M}
    end; {KeyPressed}
  end; {while}
end; {proc}

end.
```

```
unit dt2811tl;

interface

Uses Graph, Dos, Crt, vars;

const

  base   = $218;          {IO Base Address}
  ADCSR  = base + 0;           {A/D Contol/Status Register}
  ADGCR  = base + 1;           {A/D Gain/Channel}
  ADDATL = base + 2;           {A/D Low Byte}
  DADAT0L = base + 2;          {DAC0 Low Byte}
  ADDATH = base + 3;           {A/D High Byte}
  DADAT0H = base + 3;          {DAC0 High Byte}
  DADAT1L = base + 4;          {DAC1 Low Byte}
  DADAT1H = base + 5;          {DAC1 High Byte}
  DIOP0  = base + 6;           {Digital Input Port 0}
  DIOP1  = base + 6;           {Digital Output Port 1}
  TMRCRT = base = 7;           {Timer/Counter}
  DAMIN  = -5.00;              {Configured -FS Value for D/A}
  DAMAX  = 5.00;               {Configured +FS Value for D/A}
  ADMIN  = 0.00;
  ADMAX  = 5.00;

  {Bits Defined in ADCSR}
  AD_Done    = 128;            {A/D Done read Status, Set on Completion}
  AD_Error   = 64;             {A/D Error read Status, Set When Error Occurs}
  AD_Busy    = 32;             {A/D Busy read Status, Set by a Trigger}
  Clear_Error = 16;            {Write A 1 to Clear Errors}
  AD_Intr    = 4;              {Clear to Disable Interupts}
  Mode_1R    = 2;
  Mode_0R    = 1;
  Init       = Clear_Error;

  Mode_0     = 0;
  Mode_1     = 1;
  Mode_2     = 2;
  Mode_3     = 3;

  Gain_1     = 0;         {0*2**7 + 0*2**6}
  Gain_2     = 64;             {0*2**7 + 1*2**6}
  Gain_4     = 128;            {1*2**7 + 0*2**6}
  Gain_8     = 192;            {1*2**7 + 1*2**8}


var hb,lb,dac,adc:byte;
   is_error:boolean;
   Get_LIFER:Boolean;
   OutFile : Text;

   procedure MyGetTime(var time:real);
   function AtoD(adc:integer; Ggain:integer):real;
   function Din:integer;
```

```
  procedure Dout(dout:integer);
  procedure AD_Init;
  procedure DtoA(channel:integer; volt:real);
  procedure check_field;

implementation

procedure MyGetTime(var time:real);
var hour, minute, second, sec100: word;
begin
  GetTime(Hour, Minute, Second, Sec100);
  time := hour*3600.0+minute*60.0+second+sec100*0.01;
end;

function Din:integer;
begin
  Din := port[DIOP0];
end;

procedure Dout(dout:integer);
begin
  port[DIOP1] := dout;
end;


procedure DtoA(channel:integer; volt:real);
var low,high:integer; hb,lb:byte;
   setting:real;

begin

  if (channel = 0) then begin
         low := DADAT0L ;
     high := DADAT0H
    end
  else
    begin
      low := DADAT1L;
      high := DADAT1H
    end;

  {Convert from Given Voltage to One of 4096 Possibilities}
  setting := ($FFF)/(DAMax-DAMin)*(volt-DAMin);

  hb := trunc(setting/256.0);
  lb := trunc(setting - 256.0*int(setting/256.0) );

  Port[low] := lb;
  Port[high] := hb;

end;


function AtoD(adc:integer; Ggain:integer):real;
```

```
  procedure ad_set(mode, gain, channel:byte);
  var mask:byte;
  begin
    port[ADCSR] := mode;
    mask := gain + channel;
    port[ADGCR] := mask;
  end;


  function adin:real;
  var stat,lb,hb : byte;

  begin
    repeat
      stat := (Port[ADCSR]) and AD_Done;
    until stat = AD_Done;
    Lb := Port[ADDATL];
    Hb := Port[ADDATH];
    adin := admin + (hb*256+lb)*(admax-admin)/$fff;
  end;

begin
  ad_set(mode_0,Ggain,adc);
  AtoD := adin;
end;

procedure AD_Init;
var done_flag,lb,hb : byte;
begin
  Port[ADCSR] := Init;
  Lb := Port[ADDATL];
  Hb := Port[ADDATH];
end;


procedure check_field;

{
        Attempt to synchronize to magnetic field using
        microswitch ss21pe digital magnetic field sensor.

        Reverse logic.

        ss21pe output:   +5V    : No Field
                         +0V    : Field (15-25 Gauss Switch Range)

        Sensor draws power from dt2817 pin 20,
        ground is pin 22 and signal is pin 4 (port 1, bit 0)

        TJD 5/23/92


  Look for a transition from low (field present) to
  high (field absent) and mark it!
```

```
}

begin
  prev_rise := num_rise;
  mag_field := ( din and 1);
  if ( (mag_field = 1) and (prev_field = 0)) then
    num_rise := num_rise + 1;
  prev_field := mag_field;

end;


end.
```

```
Unit LIFER;

Interface

  Uses Graph, Dos, Crt, MyGraph, DT2811TL, Misc, Vars;

Const

   { CHANGED 11/3/93 TJD
   MaxLiferPoints = 25; }

   MaxLiferPoints = 10;

var
   Gr2XMin, Gr2XMax, Gr2dX, Gr2YMin, GR2YMax, Gr2dY : real;

  procedure GetLIFERPoint;
  procedure ACQ_LIFER;
  procedure StopACQ_LIFER;


Implementation

{Acquire Laser Interferometry Signal}
procedure GetLIFERPoint;

var time: array[1..MaxLiferPoints] of real;
   val : array[1..MaxLiferPoints] of real;
   sumt, sumv, tav, vav : real;
   temp, YVAL, dT0 :real;
   i:integer;s,s1:string;

begin
  sumt := 0; sumv := 0;
  for i := 1 to MaxLiferPoints do begin
    { delay(50);}             {wait 50 ms}
    MyGetTime(time[i]);  {Get Time}
    Val[i] := AToD(0,Gain_1);       {Read in A/D From Chanel 0}
    sumt := sumt + time[i];
    sumv := sumv + val[i];
  end;
  tav := sumt/MaxLiferPoints;
  vav := sumv/MaxLiferPoints;
  dT0 := tav - time0;
  LiferTMax := tav - time0;
  if (dT0) > GR2XMax then begin
    temp    := GR2XMin;
    GR2XMin := Gr2XMax;
    Gr2XMax := GR2XMax + (GR2XMax - temp);
    SetUp2(GR2XMin, GR2XMax, GR2dX, GR2YMin, GR2YMax, GR2dY);
  end;
  View2;
  if (Vav >= Gr2YMin) and (Vav <= Gr2YMax) then begin
    PutPixel(View2X(dT0), View2Y(Vav), Green);
```

```
    end;
  if (Status.Files = Open) then
    Writeln(LIFERFile, dT0:10:2, '  ',Vav:10:5);
end;

procedure ACQ_LIFER;
begin
  GR2XMin :=   0.0;
  GR2XMax := 120.0;
  GR2dX   :=  20.0;
  GR2YMin :=   0.0;
  GR2YMax :=   5.0;
  GR2dY   :=   1.0;
  SetUp2(GR2XMin, GR2XMax, GR2dX, GR2YMin, GR2YMax, GR2dY);
  Get_LIFER := True;      { Enable Data Collection }
end;

procedure StopACQ_LIFER;
var s,s1:string;
begin
  if not(Get_LIFER) then
    exit
  else begin
    Get_LIFER := False;
    SetColor(Blue);
{   str(times:6,s); s1 := 'No. Pts = '+s+' ';
    SetViewPort(v2xs, v2ys, v2xe, v2ye, ClipOn);
    outtextxy(v2xe-TextWidth(s1),4,s1);
}
  end;
end;

begin

end.
```

```
UNIT Misc;

INTERFACE
Uses Dos, Crt, AuxInOut, Vars;

PROCEDURE GetFuncKey (var Funckey : integer);
PROCEDURE GetString (var InputString : str8; KindOfString : StringKind;
                Xpos : integer; Ypos : integer);
FUNCTION  RemoveSpaces (InString : str255) : str255;
Procedure OpenOutPut;
Procedure CloseOutPut;
Procedure EnterText(var s:string; x,y:integer; max : integer);
Procedure CenterText(Y:integer; stuff:string);
Procedure FileNameGen(Var Success:Boolean);
Procedure ConvertToHex(A:Word; Var S:String);
Function  OKInput:Boolean;
procedure GetInfo;
Procedure WriteInfo;
Procedure WriteBlanks(I:Integer; var x,y : byte);
Procedure StartUp;


IMPLEMENTATION


PROCEDURE GetFuncKey (var Funckey : integer);

var
  Key : char;

begin
  Key := ReadKey;
  if Key = #0 then begin
    Key := ReadKey;
    FuncKey := ord(Key);
  end
  else begin
    Sound (300); Delay (100); NoSound;
    FuncKey := 1;
  end;
end;



PROCEDURE GetString (var InputString  : str8;
                KindOfString : StringKind;
                Xpos        : integer;
                Ypos        : integer);
                  {The procedure obtains a string from the
                  user. The string is terminasted by a return.}
var
  CharValue, BoundLow, BoundHi, LetterCount : integer;
  CharIn : char;
  NewString : string[8];
```

```
  IsReturn : boolean;

begin
  IsReturn := false;
  LetterCount := 0;
  NewString := InputString;
  LetterCount := Length(NewString);
  GotoXY(Xpos, Ypos);
  Write(NewString);

  if (KindOfString = Alfa) then begin
    BoundLow := 64; BoundHI := 91; end;
  if (KIndOfString = Numeric) then begin
    BoundLow := 44; BoundHi := 58; end;
  if (KindOfString = Mix) then begin
    BoundLow := 44; BoundHi := 91; end;

  repeat

    repeat

     {Get a character}
      CharIn := readkey;

     {Check for special code and if true read the next character}
      if (CharIn = #0) then begin
        CharIn := readkey;
        CharValue := 0;
      end

      else begin
        CharIn := UpCase(CharIn);
        CharValue := ord(CharIn);

       {Check fo termination}
        if (CharIn = chr(13)) then IsReturn := true;
        if (NewString = '') then begin
          GotoXY (Xpos,Ypos);
          Write ('        ');
          GotoXY (Xpos,Ypos);
        end;

       {Check for backspace}
        if (CharIn = chr(8)) then begin
          Delete (NewString, LetterCount, 1);
          LetterCount := LetterCount - 1;
          CharValue := 0;
          GotoXY (Xpos,Ypos);
          Write ('         ');
          GotoXY (Xpos,Ypos);
          Write (NewString);
        end;
      end;
```

```
      {Inform of a not acceptable character}
      if not ((((CharValue > BoundLow) and (CharValue < BoundHi))
        or IsReturn) or (CharValue = 32))  then begin
         Sound (300);
         Delay (200);
         NoSound;
      end;

    until (((CharValue > BoundLow) and (CharValue < BoundHi)) or IsReturn);

    if (CharIn <> chr(13)) then begin
      if ((NewString = '') and (CharIn = '.')) then NewString := '0';
      NewString := NewString+CharIn;
      LetterCount := LetterCount + 1;
      Write (CharIn);
    end;

  until IsReturn;

  if (NewString <> '') then InputString := NewString;
end;


FUNCTION RemoveSpaces (InString : str255) : str255;

begin
  while not (InString[1] in MixSet) do
    delete (InString,1,1);
  RemoveSpaces := InString;
end;


Procedure OpenOutPut;
begin
  Status.Files := Open;
  Assign(InfoFile ,Status.Path+InfoPath +Status.FileBase+'.INF');
  ReWrite(InfoFile);
  if (Status.LIFER) then begin
    Assign(LiferFile,Status.Path+LiferPath+Status.FileBase+'.LIN');
    ReWrite(LiferFile);
  end;
  if (Status.DARSS) or (Status.THR640) then begin
    Assign(OptFile  ,Status.Path+OptPath  +Status.FileBase+'.OPT');
    ReWrite(OptFile);
    if (status.TWOOPT) then begin
      Assign(Opt2File  ,Status.Path+OptPath  +Status.FileBase+'.OP2');
      ReWrite(Opt2File);
    end;
  end;
  if (Status.COMDEL) then begin
    Assign(IVFile  ,Status.Path+IVPath  +Status.FileBase+'.IV');
    ReWrite(IVFile);
    Writeln(IVFile, 'V2. COMDEL-RPM Output.');
    Writeln(IVFile, '  Time     Power  Vrf   Irf   Z   Phase  Vdc');
```

```
    Writeln(IVFile);
  end;
  if (Status.LUXTRON) then begin
    Assign(TempFile ,Status.Path+TempPath +Status.FileBase+'.LUX');
    ReWrite(TempFile);
  end;
  OutFileString := Status.FileBase;
end;


Procedure CloseOutPut;
begin
  Status.Files := Closed;
  Close(InfoFile);
  if (Status.LIFER) then  Close(LiferFile);
  if (Status.DARSS) or (Status.THR640) then  begin
    Close(OptFile);
    if (status.TWOOPT) then Close(Opt2File);
  end;
  if (Status.COMDEL) then Close(IVFile);
  if (status.LUXTRON) then Close(TempFile);
end;


Procedure EnterText(var s:string; x,y:integer; max : integer);
var loc:byte; done : boolean; ch : char;
begin
  done := false;
  loc := 0;
  GotoXY(X,Y);
  while not(done) do begin
    if Keypressed then begin
      ch := ReadKey;
      if (ch = #0) then begin
         ch := readkey;
         ch := #0;
      end;
      if ( ((ch >= 'a') and (ch <= 'z')) or ((ch >= 'A') and (ch <= 'Z'))
           or ((ch >= '0') and (ch <= '9')) or (ch = '.') or (ch = '\')
        or (ch = ':')) then begin
        if (loc < max) then begin
              inc(loc);
          s[loc] := ch;
              GotoXY(x,y); Write(Ch);
          x := x + 1;
        end {if loc}
        end {if ch....}
      else
      if (ch = ^H) then
            begin
          if (loc <> 0) then begin
            x := x - 1;
            GotoXy(X,Y);
            Write(' ');
```

```
          dec(loc);
        end {if loc<>0}
      end {if ch =^H}
    else
      if (ch = ^M) then begin
        done := tru.;
        s[0] := chr(loc);
        Write(ch);
      end; {if ch = ^M}
  end; {KeyPressed}
 end; {while}
end; {proc}


Procedure EnterInfo(var s:string; x,y:integer; max : integer);
var loc:byte; done : boolean; ch : char;
begin
  done := false;
  loc := 0;
  GotoXY(X,Y);
  while not(done) do begin
    if Keypressed then begin
      ch := ReadKey;
      if (ch = #0) then begin
        ch := readkey;
        ch := #0;
      end;
      if (ch>=#32) and (ch<=#126) then begin
        if (loc < max) then begin
              inc(loc);
          s[loc] := ch;
              GotoXY(x,y); Write(Ch);
          x := x + 1;
        end {if loc}
        end {if ch....}
      else
      if (ch = ^H) then
            begin
          if (loc <> 0) then begin
            x := x - 1;
            GotoXy(X,Y);
            Write(' ');
            dec(loc);
          end {if loc<>0}
        end {if ch =^H}
       else
        if (ch = ^M) then begin
          done := true;
          s[0] := chr(loc);
          Write(ch);
        end; {if ch = ^M}
    end; {KeyPressed}
  end; {while}
 end; {proc}
```

```
Procedure CenterText(Y:integer; stuff:string);
Var
  X : Integer;

begin
  X := (80 - Length(Stuff)) div 2;
  GotoXY(X,Y); Write(Stuff);
end;


Procedure FileNameGen(Var Success:Boolean);
{
  Generate a Unique File Name.

  IYYMDCNN

  I  = User Initial
  YY = Year - 1900 in Base 16 (1991 - 1900 = 91 = $5B)
  M  = Month in Basc 16
  D  = Day of Month in Base 36
  C  = Chamber Letter (A or B)
  NN = Run ID Number 01 -> 99

  This Subroutine examines the default path subdirectory INFO
  for .inf files until its ok.
}


Var
  Tens     : Integer;
  FileName : String;
  DirInfo  : SearchRec;     { PreDefined in DOS Unit }

begin
  Success := False;
  Status.FileBase[0] := Chr(8);
  Repeat
    Status.RunNumber := Status.RunNumber + 1;
    if (Status.RunNumber > 99) then begin
      Writeln('Run Sequence Full - Change User Initial');
      Exit;
    end;
    if (Status.RunNumber) < 10 then begin
      Status.FileBase[7] := '0';
      Status.FileBase[8] := Chr( Ord('0') + Status.RunNumber);
      end
    else begin
      Tens := Status.RunNumber div 10;
      Status.FileBase[7] := Chr( Ord('0') + Tens);
      Status.FileBase[8] := Chr( Ord('0') + Status.RunNumber - 10*Tens);
    end;
    FileName := Status.Path+InfoPath+Status.FileBase+'.INF';
    FindFirst(FileName, AnyFile, DirInfo);
  Until (DosError <> 0);
```

```
  Success := True;
end; {Procedure}


Procedure ConvertToHex(A:Word; Var S:String);
Var B0, B1, B2, B3: Integer;
   Good : Boolean;
Begin
  if (A <= 65535) and (a >= 0) then
    good := true
  else
    good := false;

  if (good) then begin
    B3 := Trunc(A/4096);
    B2 := Trunc( (A - 4096*B3)/256);
    B1 := Trunc( (A - 4096*B3 - 256*B2)/16);
    B0 := Trunc( (A - 4096*B3 - 256*B2 - 16*B1));
    S[0] := Chr(0);
    if (B3 <> 0) then begin
      Inc(S[0]);
      S[ Ord(S[0]) ] := HexChar[B3];
    end;
    if (B2 <> 0) then begin
      Inc(S[0]);
      S[ Ord(S[0]) ] := HexChar[B2];
    end;
    if (B1 <> 0) then begin
      Inc(S[0]);
      S[ Ord(S[0]) ] := HexChar[B1];
    end;
    Inc(S[0]);
    S[ Ord(S[0]) ] := HexChar[B0];
  end; {if Good}
end;

Function OKInput:Boolean;
var sum:integer;
begin
 OKInput := True;
 Sum := 0;
 if (Status.THR640PMT.WaveScan and Status.DARSS) then
   OkInput := False;
{  if (Status.THR640PMT.WaveScan) then sum := sum + 1;
  if (Status.THR640PMT.OptEmission) then sum := sum + 1;
  if (Status.THR640LDA) then sum := sum + 1;
  if (Status.DARSS) then sum := sum + 1;
}
  { More than 1 Type of OE Scan ??? }
{
  if (SUM > 1) then
    OKInput := False
  else
}
```

```
    { Trying to do Temporal and Non Temporal Stuff ??? }
    if (Status.LIFER) and (Status.THR640PMT.WaveScan) then
      OKInput := False;
    if (Status.LUXTRON) and (Status.THR640PMT.WaveScan) then
      OKInput := False;
    if (Status.DARSS) and (Status.THR640PMT.WaveScan) then
      OKInput := False;

    if (status.THR640PMT.OptEmission) and (status.DARSS) then begin
      OKInput := True;
      Status.TWOOPT := True;
    end;

end;


Procedure WriteBlanks(I:Integer; var x,y : byte);
VAR J : Integer;
Begin
  X := WhereX; Y := WhereY;
  For J := 1 to I do
    Write(' ');
  GotoXY(X,Y);
END;


procedure GetInfo;
Var S:String; X,Y :byte;
begin
  ClrScr;
  TextColor(Blue);
  Writeln('Enter Text Notes for Info File');
  Writeln('Enter a blank Line to end input');
  writeln;
  TextColor(Black);
  Y := 3;
  repeat
    Y := Y + 1;
    GotoXY(1,Y); WriteBlanks(77,x,y);
    EnterInfo(S, 1, Y, 75);
    Writeln(InfoFile, S);
  until (length(s) = 0);
end;


Procedure WriteInfo;

begin
  with Status do begin
    writeln(InfoFile, LIFER);
    writeln(InfoFile, DARSS),
    if (DARSS) then begin
      writeln(InfoFile, DARSSParam.Grating, ' ',
                        DARSSParam.Center:10:4);
```

```
      writeln(InfoFile, DARSSParam.Exposure, ' ',
                        DARSSParam.Correction:10:4);
    end; { if DARSS }
    writeln(InfoFile, Comdel);
    writeln(InfoFile, THR640);
    if (THR640) then begin
      writeln(InfoFile, Grating);
      writeln(InfoFile, InputSlit:5:1);
      writeln(InfoFile, OutputSlit:5:1);
      writeln(InfoFile, LDA);
      writeln(InfoFile, PMT);
      if (PMT) then begin
        writeln(InfoFile,Status.PMTVoltage:6:0);

        { Added 1/7/94 }

        writeln(InfoFile, Status.THR640PMT.Detector:6);
        writeln(InfoFile, Status.THR640PMT.Gain:6);
        writeln(InfoFile, Status.THR640PMT.Gain2:6);
        writeln(InfoFile, Status.THR640PMT.RiseTime:6);

        writeln(InfoFile,THR640PMT.OptEmission);
        if (THR640PMT.OptEmission) then
          writeln(InfoFile, Status.StartingWaveLength:7:1);
        writeln(InfoFile,THR640PMT.WaveScan);
        if (THR640PMT.WaveScan) then Begin
          writeln(InfoFile, Status.StartingWaveLength:7:1);
          writeln(InfoFile, Status.EndingWaveLength:7:1);
          writeln(InfoFile, Status.WaveLengthStep:7:4);
          writeln(InfoFile, Status.KeitAvg:3);
            end; { if WaveScan }
      end; { if PMT }
    end; { If THR640 }
    writeln(InfoFile, Path);
    writeln(InfoFile, FileBase);
    writeln(InfoFile, LUXTRON);
  end; { With Status }
  writeln(InfoFile, '***** BEGIN RUN DATA *****');
end; { WriteInfo }


Procedure StartUp;
Var
  Ch:String; Cha : Char;
  X,Y : Byte;
  Year, Month, Day, DayOfWeek : Word;
  Good,done,Happy : Boolean;
  I : Integer;
begin
  Happy := false;
  Repeat
  WITH Status Do Begin
    RunNumber := 0;
    Files := Closed;
```

```
TextBackGround(LightGray);
ClrScr;
TextColor(Red+Blink);
CenterText(1,'A.M.E. 5000  Automated Data Acquisition System');
TextColor(Blue);
GotoXY(1,4); Write('Enter User Initial : ');
TextColor(Black);
WriteBlanks(3,x,y);
EnterText(Ch, X, Y, 1);
FileBase[1] := UpCase(Ch[1]);
GotoXY(X,Y); Write(FileBase[1]);
GetDate(Year, Month, Day, DayOfWeek);
ConvertToHex(Year-1900, Ch);
If (Ord(Ch[0]) = 1) then
  begin
    FileBase[2] := '0';
    FileBase[3] := Ch[1];
  end
else
  begin
    FileBase[2] := Ch[1];
    FileBase[3] := Ch[2];
  end;
FileBase[4] := HexChar[ Trunc(Month) ];
FileBase[5] := DblHexChar[ Trunc(Day) ];
Good := False;
Repeat
  TextColor(Blue);
  GotoXY(1,6); Write('Enter Chamber Initial (A/B) : ');
  TextColor(Black);
  WriteBlanks(3,x,y);
  EnterText(Ch, X, Y, 1);
  FileBase[6] := UpCase(Ch[1]);
  If (FileBase[6]= 'A') or (FileBase[6] = 'B') then
    begin
      Good := True;
      GotoXY(X,Y);
      Write(FileBase[6]);
    end
  Else
    Good := False;
Until Good;
FileBase[0] := Chr(6);
TextColor(Blue);
GotoXY(1,8); Write('File Name Base : ');
TextColor(Black);  Write(FileBase);
TextColor(Blue);

GotoXY(1,10); Write('Enter File Path Name : ');
TextColor(Black);
If (FileBase[1] = 'T') then begin
  PATH := 'F:\TIM\EXPERIME\';
  X := WhereX;
  Y := WhereY
```

```
    end
  Else begin
    WriteBlanks(42,x,y);
    EnterText(Ch, X, Y, 40);
    For I := 1 to Ord(Ch[0]) Do
      Path[I] := UpCase(Ch[I]);
    Path[0] := Ch[0];
    If (Path[ Ord(Path[0]) ] <> '\') then
      begin
        Inc( Path[0] );
        Path[ Ord(Path[0]) ] := '\';
      end;
  end;
  GotoXY(X,Y);  Write(Path);
  TextColor(red+blink);
  CenterText(13,'A C Q U I S I T I O N   P A R A M E T E R S');
  TextColor(Blue);
  GotoXY(5,15); write('Laser InterFerometry   : :');
  GotoXY(5,16); write('THR640 WaveLength Scan  : :');
  GotoXY(5,17); write('THR640 Optical Emission : :');
  GotoXY(5,18); write('Luxtron Temperature    : :');
  GotoXY(45,15); Write('THR640 Diode Array     : :');
  GotoXY(45,16); Write('DARSS Diode Array      : :');
  GotoXY(45,17); Write('Comdel RPM             : :');
  GotoXY(5,19);
  write('Press X to Select/Deselect, Up/Down Arrow to Move, Return When Done');
  TextBackground(Cyan);
  GotoXY(30,15); write(' ');
  GotoXY(30,16); write(' ');
  GotoXY(30,17); write(' ');
  GotoXY(30,18); write(' ');
  GotoXY(70,15); write(' ');
  GotoXY(70,16); write(' ');
  GotoXY(70,17); write(' ');
  X := 30; Y := 15;
  done := false;
  { Remember, We're still inside a with do }
  Lifer                      := False;
  THR640PMT.WaveScan         := False;
  THR640PMT.OptEmission      := False;
  THR640LDA                            := False;
  DARSS                      := False;
  COMDEL                     := False;
  LUXTRON                    := False;
  TWOOPT                     := False;
  Repeat
   GotoXY(X,Y); TextColor(red);
       if (x=30) and (y=15) Then
         if LIFER then
           write('X')
    else
           Write(' ');
   if (x=30) and (y=16) then
         if THR640PMT.WaveScan then
```

```
        write('X')
  else
        Write(' ');
if (x=30) and (y=17) then
  if THR640PMT.OptEmission then
        write('X')
  else
        Write(' ');
if (x=30) and (y=18) then
  if LUXTRON then
        write('X')
  else
        Write(' ');
if (x=70) and (y=15) then
  if THR640LDA then
        write('X')
  else
        Write(' ');
if (x=70) and (y=16) then
  if DARSS then
        write('X')
  else
        Write(' ');
if (x=70) and (y=17) then
  if COMDEL then
        write('X')
  else
        Write(' ');
GotoXY(X,Y);
Repeat Until KeyPressed;
cha := ReadKey; Cha := UpCase(Cha);
if (Cha = 'X') then begin
  case (X+Y) of
    45: begin
          GotoXY(X,Y); TextColor(Red);
          if Lifer then begin
            lifer := false;
            write(' ');
            end
          else begin
            lifer := true;
            write('X');
          end;
          Y := Y + 1;
        end;
    46: begin
        GotoXY(X,Y); TextColor(Red);
          if THR640PMT.WaveScan then begin
            THR640PMT.WaveScan := false;
            write(' ');
            end
          else begin
            THR640PMT.WaveScan := true;
            write('X');
```

```
      end;
      Y := Y + 1;
    end;
  47: begin
    GotoXY(X,Y); TextColor(Red);
      if THR640PMT.OptEmission then begin
        THR640PMT.OptEmission := false;
        write(' ');
        end
      else begin
        THR640PMT.OptEmission := true;
        write('X');
      end;
      Y := Y + 1;
    end;
  48: begin
    GotoXY(X,Y); TextColor(Red);
      if LUXTRON then begin
        LUXTRON := false;
        write(' ');
        end
      else begin
        LUXTRON := true;
        write('X');
      end;
      X := 70; Y := 15;
    end;

  85: begin
    GotoXY(X,Y); TextColor(Red);
      if THR640LDA then begin
        THR640LDA := false;
        write(' ');
        end
      else begin
        THR640LDA := true;
        write('X');
      end;
      Y := Y + 1;
    end;
  86: begin
    GotoXY(X,Y); TextColor(Red);
      if DARSS then begin
        DARSS := false;
        write(' ');
        end
      else begin
        DARSS := true;
        write('X');
      end;
      Y := Y + 1;
    end;
  87: begin
    GotoXY(X,Y); TextColor(Red);
```

```
          if COMDEL then begin
            COMDEL := false;
            write(' ');
            end
          else begin
            COMDEL := true;
            write('X');
          end;
          X := 30; Y := 15;
        end;
   end; { CASE }
 end; {if X}
 if ( (ORD(Cha) = 13) and (OKInput)) then done := true;
 if cha = #0 then begin
        cha := ReadKey;
    case ord(cha) of
      72 : begin {Up}
          Y := Y - 1;
          if (X = 30) and (y < 15) then begin
            X := 70; Y := 17; end
          else
            if (x = 70) and (y < 15) then begin
              X := 30; Y := 18; end;
          end;
      80 : begin {DOWN}
          Y := Y + 1;
          if (X=30) and (Y > 18) then begin
            X := 70; Y := 15; end
          else
            if (x=70) and (y>17) then begin
              X := 30; Y := 15; end;
          end;
    end; {case}
 end; {if ch=0}
Until Done;
PMT := false; LDA := False; THR640 := False;
if (THR640PMT.WaveScan) or (THR640PMT.OptEmission) then
  begin
    PMT := True;
    THR640 := True;
  end;
if (THR640LDA) then begin
  LDA := True;
  THR640 := True;
end;
{ Should We Read Input for The Grating ? }
if (thr640) then
  Good := False
else
  Good := True;
TextBackGround(LightGray);
GotoXY(1,19); for i := 1 to 74 do write(' ');
While not(good) do begin
  TextColor(Blue);
```

```
  GotoXY(1,19);
  writeln('THR-640  Diffraction Gratings Available: ');
  TextColor(Red);
  writeln('A: 4500 - 10000 Ang.  1800 g/mm');
  writeln('B: 2500 Ang. Blaze    1800 g/mm'); writeln;
  TextColor(Blue);
      Write('Enter Grating to Use (A/B) : ');
  TextColor(Black);
  WriteBlanks(3,x,y);
  EnterText(Ch, X, Y, 1);
  Grating := UpCase(Ch[1]);
  If (Grating = 'A') or (Grating = 'B') then
    begin
      Good := True;
      GotoXY(X,Y);
      Write(Grating);
    end
  Else
    Good := False;
end; {While}
{ Should We Read Input for The DARSS Grating ? }
if (DARSS) then
  Good := False
else
  Good := True;
TextBackGround(LightGray);
GotoXY(1,19); for i := 1 to 74 do write(' ');
While not(good) do begin
  TextColor(Blue);
  GotoXY(1,19); Write('Enter the Grating (1-3) on the Jarrel-Ash : ');
  TextColor(Black);
  WriteBlanks(3,x,y);
  EnterText(Ch, X, Y, 1);
  DARSSParam.Grating := ord(Ch[1]) - ord('0');
  Ch[1] := UpCase(Ch[1]);
  If (Ch[1] = '1') or (Ch[1] = '2') or (Ch[1] = '3') then
    begin
      Good := True;
      GotoXY(X,Y);
      Write(Ch[1]);
    end
  Else
    Good := False;
  GotoXY(1,20);
  TextColor(Blue);
  Write('Enter Meter Reading from Jarrel-Ash in Angstroms : ');
  TextColor(Black);
  WriteBlanks(10,x,y);
      Readln(DARSSParam.Center);
  TextColor(Blue);
  Write('Enter Exposure Time in ms : ');
  TextColor(Black);
  WriteBlanks(10,x,y);
      Readln(DARSSParam.Exposure);
```

```
      TextColor(Blue);
      Write('Enter Correction Factor in Angstroms : ');
      TextColor(Black);
      WriteBlanks(10,x,y);
          Readln(DARSSParam.Correction);
    end; {While}
  end; {WITH}
    Repeat
      TextColor(Blue);
      GotoXY(1,24); Write('Are All These Correct? (Y/N) : ');
      TextColor(Black);
      WriteBlanks(3,x,y);
      EnterText(Ch, X, Y, 1);
      Cha := UpCase(Ch[1]);
      If (Cha = 'Y') or (Cha = 'N') then
        begin
          Good := True;
          GotoXY(X,Y);
          Write(Cha);
        end
      Else
        Good := False;
    Until Good;
    if (Cha = 'Y') then
      Happy := true
    else
      Happy := False;
  Until Happy;
{
  if (Status.files = closed) then begin
    FileNameGen(Happy);
    OpenOutPut;
  end;
  Writeln(InfoFile,'***** Begin Pre-Run Comments *****');
  GetInfo;
  Writeln(InfoFile,'***** End Pre-Run Comments *****');
  WriteInfo;
}
end;


begin
end.
```

```
UNIT Keit;

INTERFACE
uses DOS, Crt, Misc,vars;

CONST

  EOICode=' EOI';              {End or Identifiy Terminator Code}
  BufferedOutput='BUFFERED';       {IOTech 488 Buffered Data Transfer Code}
                           {Number Characters}

VAR
  Response         : Str255;             {IOTech Response}

PROCEDURE CloseKeithley;
PROCEDURE CloseComm;
PROCEDURE InitCommun;
PROCEDURE InitKeithley;
PROCEDURE IsKeithleyReady;
FUNCTION InputKeithley(NumReadingsPerPt: Integer) : double;

IMPLEMENTATION

PROCEDURE CloseKeithley;

begin
 {Terminate comunication via IOTech card}
  Writeln (IeeeOut, 'CLEAR 22');
  Writeln (IeeeOut, 'LOCAL 22');
  { Writeln (IeeeOut, 'RESET'); }
end;

PROCEDURE CloseComm;
begin
  Close (IeeeOut);
  Close (IeeeIn);
end;

PROCEDURE InitCommun;

  {Initialize the IEEE Communications and check to make
   sure instruments connected to the IOTech card are working.}

var
  Response : str255;

begin
  Assign (IeeeOut, 'IeeeOut');
  Rewrite (IeceOut);
  Assign (IeeeIn, 'IeeeIn');
  Reset (IeeeIn);

  {Initialize Status Request Line}
  Writeln(IeeeOut,'ARM SRQ');
```

```
    {Requests IOTech 488 identity}
    {GotoXY (2,10);}
    Writeln ('Testing Communication:');
    Writeln (IeeeOut, 'RESET');
    Writeln (IeeeOut, 'HELLO');
    Readln (IeeeIn, Response);
    Response := RemoveSpaces(Response);
    {GotoXY (2,11);}
    Writeln (Response);

    if (status.thr640pmt.detector = 0) then begin
       {Request Keithley 485 to identify}
       Writeln (IeeeOut, 'OUTPUT 22;U0X');
       Writeln (IeeeOut, 'ENTER 22');
       Readln (IeeeIn, Response);
       Response := RemoveSpaces(Response);
       {GotoXY (2,12);}
       writeln;
       Writeln('   Device --- ', Response, '   OK');
       Writeln;
    end; { if status.thr640pmt.detector }
    Writeln (IeeeOut, 'RESET');
    Writeln (IeeeOut, 'LOCAL');
    writeln;
end;

PROCEDURE InitKeithley;
{Initialize IEEE communication to the Keithley ammeter}

Begin
  Writeln (IeeeOut, 'REMOTE 22'); {Enable computer control of all devices}
  Writeln (IeeeOut, 'ARM SRQ');   {Enable SRQ to set light-pen interupt}

              {Send the following commands to the Keithley 485
                 C0  : zero check off
                 D0  : LOG off
                 R0  : auto range control
                 Z0  : REL off
                 K0  : EOI (end or identify) enabled
                 G1  : readings without prefix (data format)
                 M9  : Reading done or reading overflow generates SRQ
                 T5  : trigger is one-shot on X}
  Writeln (IeeeOut, 'OUTPUT 22;C0D0R0Z0K0X');
  Writeln (IeeeOut, 'OUTPUT 22;G1M9X');
  Writeln (IeeeOut, 'OUTPUT 22;T0X');

  {Pause for Keithley to set SRQ mask}
  Delay (100);

  Writeln (IeeeOut, 'OUTPUT 22;X');
  Writeln (IeeeOut, 'ENTER 22');
  Readln (IeeeIn, Response);
end;
```

```
PROCEDURE IsKeithleyReady;

Var
  K485       : integer;
  IsInterrupt : boolean;

begin
  Writeln (IeeeOut, 'OUTPUT 22;X');
  K485 := 0;

  {Check for interrupt and determine the cause}
  IsInterrupt := false;
  repeat
    Writeln (IeeeOut, 'SPOLL 22');
    Readln (IeeeIn, K485);
    if (K485 > 63) and ((K485 and 32) <> 1 ) then begin
      if (K485 and 8) <> 0 then IsInterrupt := true;
      if (K485 and 0) <> 0 then begin
        GotoXY (5,24); Write ('ERROR: Reading Overflow');
      end;
    end;

    if (K485 and 32) <> 0 then begin
      GotoXY (5,24);
      Write ('ERROR');
      Writeln (IeeeOut, 'OUTPUT 22;U0X');
      Writeln (IeeeOut, 'ENTER 22');
      Readln (IeeeIn, Response);
      Response := RemoveSpaces(Response);
      GotoXY (10,25); Write ('  ', Response);
    end;
{     if keypressed then begin
      GetFuncKey (FuncKey);
      if FuncKey = 68 then IsInterrupt := true;
    end;
}  until (IsInterrupt);
end;

FUNCTION InputKeithley(NumReadingsPerPt: Integer) : double;

{Read data from the PMT via the Keithley 485: averaged over time}

VAR
   tempY,Y : double;
   J,code  : Integer;

begin

  J := 0;
  tempY := 0;

  repeat
    {Check if data from Keithley 485 is ready}
    IsKeithleyReady;
```

```
    {Get data}
    Writeln (IeeeOut, 'ENTER 22');
    Readln (IeeeIn, Response);

    {Determine the real value of the data from the response}
    Response := RemoveSpaces (Response);
    Val (Response, Y, code);
    tempY := tempY + Y;
    J := J + 1;
  until (J = NumReadingsPerPt);

  InputKeithley := tempY/J;

end;

begin
end.
Unit HR640;

{   This unit contains procedures used to control the SpectraLink
    to take optical spectra

    J. Liu -- March 1991
    Tim Dalton April, 1991 : Modified for AME5000

}

INTERFACE

Uses Graph, Dos, Crt, AuxInOut, Misc, Vars,
   DT2811TL, Keit, MyGraph;

TYPE

  WaveScanRec = Record
    StartingWaveLength    : Real;
    EndingWaveLength      : Real;
    WaveLengthStep        : Real;
  end;

  DirType = (Forward, Reverse);


CONST

   { Board Addresses }
   MDR : Byte = 3;
   HV  : Byte = 4;
   MCC : Byte = 9;
   PIO : Byte = 10;

   { Commands }

   BusyTest    : Byte = 0;
```

```
SelectMotor1 : Byte = 49;
SelectMotor2 : Byte = 50;
WakeUp        : Byte = 58;
DeSelect    : Byte = 58;
ReceiveData  : Byte = 63;
LoadAbsolute : Byte = 65;
Stop    : Byte = 66;
LoadCounter  : Byte = 67;
GoForward    : Byte = 70;
Go              : Byte = 71;
GoHome       : Byte = 72;
GoReverse    : Byte = 78;
LoadSpeed    : Byte = 83;
LoadTarget   : Byte = 84;
LoadVoltage  : Byte = 86;
ReadAbsolute : Byte = 97;
ReadID          : Byte = 105;

{ Other Constants }

BusyOK     : Byte = 98;          { b }
Busy       : Byte = 66;   { B }
TermKey    : Byte = 68;   { F10 }

{ Monochromator Constants }

StepsPerAngstrom : Integer = 150;

{ Grating used is 1800 g/mm, reference is 1200 g/mm
  So conversion is 1800/1200 }
GratingGrooveDensity : Real = 1.5;
ScanSpeed : Integer = 2;

ExitMirror   : Integer = 2;
GratingTurret : Integer = 1;

LateralExit  : DirType = Forward;
AxialExit    : DirType = Reverse;
GratingA     : DirType = Reverse;
GratingB     : DirType = Forward;


VAR
  Com1In, Com1Out       : Text;
  TextIn, TextOut       : char;
  NumOut                : word;
  Speed                 : Integer;
{  WaveScan             : WaveScanRec; }
  Meter                 : Real;
  WaveLength            : Real;
  Terminate             : Boolean;
  Resp          : Char;
```

```
Procedure InitCom1Port;
Procedure CloseCom1Port;
Procedure WakeUpSpectraLink(ModuleAddress:byte; var Terminate:boolean);
Procedure ReadFromSpectraLink(Var NumOut:word);
Procedure SendToSpectraLink(NumIn:word; var Terminate:boolean;
                var NumOut:word);
Procedure IsPortBusy (ModuleAddress:byte; var Terminate:boolean);
Procedure ConvertToByte (Number:longint; var LSB:word;
              var ISB:word; var MSB:word);
Procedure ConvertByteToNumber (var Number:longint; LSB,ISB,MSB:word);
Procedure LoadAbsolutePosition (wavelength:real; StepsPerAngstrom:integer);
Procedure SetSpeed(Speed:Integer);
Procedure LoadTargetPosition (Wavelength:real; StepsPerAngstrom:integer);
Procedure MoveGrating(var Terminate:Boolean);
Procedure StopGrating;
Procedure UpDateWavelength (wavelength:real; GratingGrooveDensity:real);
Procedure ReadAbsolutePosition(Var WaveLength:Real; StepsPerAngstom:integer);
Procedure SetVoltage(Volts:Real; var Done:Boolean);
Procedure HVOff(var Terminate:Boolean);
Procedure TensionCable;
Procedure MCCMotorMove(MotorNumber : Integer; Direction : DirType);
Procedure GetWaveScanParam;
Procedure GetStartingWave;
Procedure ReadMeter;
Procedure GetSpeed;
Procedure MoveToStartingPosition;
Procedure GetVolt;
Procedure GetKeitAvg;
Procedure GetSlit;
Procedure ControlTHR640(Var Terminate:Boolean);


IMPLEMENTATION

Procedure InitCom1Port;
Begin
{
  $A3 = 10100011,          $E3 = 11100011
  Baud Rate  = 2400 = 101      = 9600 = 111
  Parity    = None =  X0
  StopBit   = 0   =  0
  WordLength = 8   =  11

  From IBM Tech Ref. Manual, Pg. 5-43, Interupt 14, RS232_IO
}
  AssignAux (Com1In,1,$A3);
  AssignAux (Com1Out,1,$A3);
  ReWrite (Com1In);
  ReSet (Com1Out);
end;

Procedure CloseCom1Port;
Begin
  Close(Com1In);
```

```
  Close(Com1Out);
 end;

 Procedure ReadFromSpectraLink(Var NumOut:word);

 Var
   TextOut : Char;

 begin
     Delay(10);
     Write(Com1In, ReceiveData);
     Delay(10);
     Read (Com1Out, TextOut);
     NumOut := ord (TextOut);
 end;


Procedure SendToSpectraLink(NumIn:word; var Terminate:boolean;
                 var NumOut:word);

Var
  TextIn    : char;
  TextOut   : char;
  Funckey   : integer;

Begin
  TextIn := chr(NumIn);
  Write (Com1In, TextIn);
  Terminate := false;
  repeat
    Read (Com1Out, TextOut);
    NumOut := ord (TextOut);
    If keypressed then begin
      GetFunckey (FuncKey);
      If Funckey = TermKey then Terminate := true;
    end;
  until ((NumOut = NumIn) or Terminate);
end;


Procedure WakeUpSpectraLink(ModuleAddress:byte; var Terminate:boolean);

Begin
  {Send wake call to Spectra Link}
  SendToSpectraLink(WakeUp, Terminate, NumOut);
  {Check Module Address}
  SendToSpectraLink(ModuleAddress, Terminate, NumOut);
end;



Procedure IsPortBusy (ModuleAddress:byte; var Terminate:boolean);

Var
```

```
  Funckey  : integer;
  Problem  : boolean;

Begin
  WakeUpSpectraLink (ModuleAddress, Terminate);

  repeat
    If not(Terminate) then SendToSpectraLink(BusyTest,Terminate,NumOut);

    {Number of bytes to follow}
    If not(Terminate) then SendToSpectraLink(0,Terminate,NumOut);

    If not(Terminate) then begin
      write(Com1In, ReceiveData);
      Terminate := false;
      repeat
        Read(Com1Out, TextOut);
        NumOut := ord(TextOut);
        If keypressed then begin
          GetFunckey(Funckey);
          If Funckey = TermKey then Terminate := true;
        end;
      until (Terminate or (NumOut = Busy) or (NumOut = BusyOK));
    end;
  until (Terminate or (NumOut = BusyOK));

  {Release Spectra Link}
  SendToSpectraLink(DeSelect,Problem,NumOut);
end;


Procedure ConvertToByte (Number:longint; var LSB:word;
                 var ISB:word; var MSB:word);

var a:word;
{$R-}
Begin
  MSB := Trunc(Number/65536);
{ writeln('MSB = ',msb); }
  a := number - msb*65536;
{ writeln('a = ',a);    }
  ISB := Trunc(a/256);
{ writeln('ISB = ',isb); }
  LSB := a-ISB*256;
{ writeln('lSB = ',lsb); }
{$R+}
end;


Procedure ConvertByteToNumber (var Number:longint; LSB,ISB,MSB:word);
{used to convert hexadecimals to decimals}
Begin
  Number := LSB + ISB*256 + MSB*65536;
end;
```

```
Procedure LoadAbsolutePosition (wavelength:real; StepsPerAngstrom:integer);
{The wavelength reading must be converted to steps by first multiplying
by the number of steps per unit then converted into three bytes
to send to the MDR module of the Spectra Link.
      Grating of 1800 grooves/mm ---> 150 steps/A
         "    " 2400 grooves/mm ---> 200 steps/A}

Var
  Number     : longint;
  Terminate  : boolean;
  LSB,ISB,MSB : word;

Begin
  Number := Round(Wavelength*StepsPerAngstrom);
  ConvertToByte(Number,LSB,ISB,MSB);
  WakeUpSpectraLink (MDR, Terminate);

  {Load Absolute Position Code}
  If not(Terminate) then SendToSpectraLink(LoadAbsolute,Terminate,NumOut);

  {Number of Bytes to follow = 3}
  If not(terminate) then SendToSpectraLink(3,Terminate,NumOut);

  {Send Position in 3 Bytes}
  If not(Terminate) then SendToSpectraLink(LSB,Terminate,NumOut);
  Delay (10);

  If not(Terminate) then SendToSpectraLink(ISB,Terminate,NumOut);
  Delay(10);

  If not(Terminate) then SendToSpectraLink (MSB,Terminate,NumOut);
  Delay(10);

  {Reset Spectra Link before termination}
  SendToSpectraLink(DeSelect,Terminate,NumOut);
  If Terminate then begin
    GotoXY(5,23); write('Spectra Link not released after LOAD ABS. POS. ');
  end;
end;


Procedure SetSpeed(Speed:integer);

{ The speed parameter may be from 1 to 65535,
 where 1 is the fastest speed corresponding to 1/0.000150 = 6666 Hz
     2 corresponds to 3333 Hz and
 and 65535 is the slowest.
 A speed of 10 will work with all ISA monochromators. }

Var
  Terminate : boolean;
  LSB,MSB   : word;
```

```
Begin
  WakeUpSpectraLink (MDR, Terminate);

  MSB := Trunc(Speed/256);
  LSB := Speed - 256*MSB;

  If not(Terminate) then SendToSpectraLink (LoadSpeed,Terminate,NumOut);

  {Number of bytes to follow}
  If not(Terminate) then SendToSpectraLink (2,Terminate,NumOut);

  If not(Terminate) then begin
    SendToSpectraLink (LSB,Terminate,NumOut);
    SendToSpectraLink (MSB,Terminate,NumOut);
  end;

  SendToSpectraLink(DeSelect,Terminate,NumOut);
  If Terminate then begin
    GotoXY(5,23); write('Spectra Link not released after LOAD Speed   ');
  end;
end;


Procedure LoadTargetPosition (Wavelength:real; StepsPerAngstrom:integer);
{The ending wavelength of the monochromator, in nanometers, must be
multiplied by steps per unit and then converted into bytes}

Var
  MSB,ISB,LSB : word;
  Number      : longint;
  Terminate   : boolean;

Begin
  Number := Round(Wavelength*StepsPerAngstrom);
  ConvertToByte (Number,LSB,ISB,MSB);

  WakeUpSpectraLink (MDR, Terminate);

  {Load target position}
  If not(Terminate) then SendToSpectraLink (LoadTarget,Terminate,NumOut);

  If not(Terminate) then begin
    {Number of bytes to follow}
    SendToSpectraLink (3,Terminate,NumOut);

    {Send LSB}
    SendToSpectraLink (LSB,Terminate,NumOut);
    Delay(10);              {Delays are added to make sure the Spectra}
                   {is ready for next number}
    {Send ISB}
    SendToSpectraLink (ISB,Terminate,NumOut);
    Delay(10);

    {Send MSB}
```

```
      SendToSpectraLink (MSB,Terminate,NumOut);
      Delay(10);
    end;

    SendToSpectraLink(DeSelect,Terminate,NumOut);
    If Terminate then begin
      GotoXY(5,23);  write('Spectra Link not released after TARGET POSITION');
    end;
  end;

  Procedure MoveGrating(var Terminate:Boolean);
  {The MDR of the Spectra Link needs the Absolute and the Target Position
   before this procedure is activated}

  Begin
    WakeUpSpectraLink (MDR, Terminate);
    If not(Terminate) then SendToSpectraLink (Go,Terminate,NumOut);
    If not(Terminate) then SendToSpectraLInk (0,Terminate,NumOut);
    If not(Terminate) then SendToSpectraLink (DeSelect,Terminate,NumOut);
  end;

  Procedure StopGrating;

  Var
    Terminate   : boolean;

  Begin
    WakeUpSpectraLink (MDR, Terminate);
    SendToSpectraLink (Stop,Terminate,NumOut);
    SendToSpectraLink (0,Terminate,NumOut);
    ReadFromSpectraLink(NumOut);
    SendToSpectraLink (DeSelect,Terminate,NumOut);
    If Terminate then begin
      GotoXY(5,23);  write('Spectra Link not released after STOP Grating');
    end;
  end;


Procedure UpDateWavelength (wavelength:real; GratingGrooveDensity:real);

var
  meter  : real;

begin
  meter := wavelength * GratingGrooveDensity;
  GotoXY(68,14);  Write (Meter:8:1);
end;


Procedure ReadAbsolutePosition(Var WaveLength:Real; StepsPerAngstom:integer);

Var
  Number     : longint;
  Terminate  : boolean;
```

```
  LSB,ISB,MSB : word;

 Begin
  WakeUpSpectraLink (MDR, Terminate);

  {Load Absolute Position Code}
  If not(Terminate) then SendToSpectraLink(ReadAbsolute,Terminate,NumOut);

  {Number of Bytes to follow = 3}
  If not(terminate) then SendToSpectraLink(3,Terminate,NumOut);

  {Send Position in 3 Bytes}
  ReadFromSpectraLink(LSB);
  ReadFromSpectraLink(ISB);
  ReadFromSpectraLink(MSB);

  ConvertByteToNumber(Number, LSB, ISB, MSB);

  WaveLength := Number / StepsPerAngstrom;

  {Reset Spectra Link before termination}
  SendToSpectraLink(DeSelect,Terminate,NumOut);
  If Terminate then begin
    GotoXY(5,23); write('Spectra Link not released after READ ABS. POS. ');
  end;
end;


Procedure SetVoltage(Volts:Real; var Done:Boolean);
Var LSB,MSB   : word;  Terminate:Boolean;

Begin
 if ( (Volts >= 0) and (Volts <= 2000) ) then
   Done := False
 else
   Done := True;
 if not(Done) then begin
   WakeUpSpectraLink(HV, Terminate);
   if (terminate) then
    write('Spectra Link Not Responding to HV Wake Up');
   Volts := Volts * 2;      { Multiply By 2 }
   MSB := Trunc(Volts/256);
   LSB := Trunc(Volts) - 256*MSB;
   SendToSpectraLink(LoadVoltage, Terminate, NumOut);
   SendToSpectraLink(2, Terminate, NumOut);
   SendToSpectraLink(LSB, Terminate, NumOut);
   SendToSpectraLink(MSB, Terminate, NumOut);
   SendToSpectraLink(DeSelect, Terminate, NumOut);
   if (terminate) then begin
     write('Spectra Link not released after SetVoltage');
   end;
 end;
end;
```

```
Procedure HVOff(var Terminate:Boolean);

Begin
  WakeUpSpectraLink(HV, Terminate);
  SendToSpectraLink(Stop, Terminate, NumOut);
  SendToSpectraLink(0, Terminate, NumOut);
  ReadFromSpectraLink(NumOut);
  SendToSpectraLink(DeSelect, Terminate, NumOut);
  if (terminate) then begin
    write('Spectra Link not released after HVOff');
  end;
end;

procedure TensionCable;

{
  Retension Cable Between Grating Drive Gear Train and Mechanial
  Counter
}

Var
  Terminate : Boolean;

begin
  { As we desire to Scan Up, We Should move the meter Down and then
    back up so that the slack in the cable is removed in the Up
    direction }

  LoadAbsolutePosition(200, 1);
  SetSpeed(10);
  LoadTargetPosition(0, 1);
  MoveGrating(Terminate);
  IsPortBusy(MDR, Terminate);
  LoadTargetPosition(200, 1);
  MoveGrating(Terminate);
  IsPortBusy(MDR, Terminate);

end;

Procedure MCCMotorMove(MotorNumber : Integer; Direction : DirType);

{
  This Procedure Moves the MCC Controlled Stepping Motors

  Motor 1: Grating Turret
    1540 Steps is Full Range
    1540 = 4,6
    Use 150,6 = 1686 Steps

    GoForward -> Grating B
    GoReverse -> Grating A

  Motor 2: Exit Mirror
    220 Steps if the Full Range
```

```
    220 = 220,0
    Use 512 = 0,2 as a safety to insure complete mirror positioning
    This has been ok'ed and recommended by Roger at ISA in a
    phone conversation during the week of May 6, 1991

    GoForward -> Exit is Lateral (PMT)
    GoReverse -> Exit is Axial (LDA)

}

VAR
  LSB, MSB : Byte;
  DirCommand : Byte;
  MotCommand : Byte;
  Terminate : Boolean;

Begin

  If (MotorNumber = 1) then
    begin
      MotCommand := SelectMotor1;
      LSB := 100;
      MSB := 6;
    end
  else
    begin
      MotCommand := SelectMotor2;
      LSB := 0;
      MSB := 2;
    end;

  If (Direction = Forward) then
    DirCommand := GoForward
  else
    DirCommand := GoReverse;

  WakeUpSpectraLink(MCC, Terminate);
  SendToSpectraLink(MotCommand, Terminate, NumOut);
  SendToSpectraLink(0, Terminate, NumOut);
  SendToSpectraLink(LoadCounter, Terminate, Numout);
  SendToSpectraLink(2, Terminate, NumOut);
  SendToSpectraLink(LSB, Terminate, NumOut);
  SendToSpectraLink(MSB, Terminate, Numout);
  SendToSpectraLink(DirCommand, Terminate, NumOut);
  SendToSpectraLink(0, Terminate, NumOut);
  SendToSpectraLink(DeSelect, Terminate, NumOut);
  IsPortBusy(MCC, Terminate);
end; { Procedure }


Procedure GetWaveScanParam;
begin
  Write('Enter Starting WaveLength : ');
  Readln(Status.StartingWaveLength);
```

```
   Write('Enter Ending WaveLength : ');
   Readln(Status.EndingWaveLength);
   Write('Enter Step : ');
   Readln(Status.WaveLengthStep);
 end;


 Procedure GetStartingWave;
 begin
   Write('Enter WaveLength :');
   Readln(Status.StartingWaveLength);
 end;

 Procedure ReadMeter;

 begin
   writeln;write('Enter Meter Reading : ');
   readln(Meter);
   writeln('Loading Absolute Position');
   WaveLength := Meter / GratingGrooveDensity;
   LoadAbsolutePosition(WaveLength, StepsPerAngstrom);
   ReadAbsolutePosition(WaveLength, StepsPerAngstrom);
   Writeln('Absolute Position = ',wavelength:8:3,' A');
   Writeln('Meter Reading     = ',wavelength*GratingGrooveDensity:8:3,' A');
 end;


Procedure GetSpeed;
begin
  Writeln('Loading Speed');
  Terminate := False;
  repeat
    writeln; write('Enter Speed 1-65535: ');
    readln(speed);
    if ((speed <= 65535) and (speed >= 1)) then terminate := true;
  until terminate;
  SetSpeed(Speed);
end;


Procedure MoveToStartingPosition;

Var
   Done : Boolean;
   abspos : real;

Begin
  { Move to Initial Position }
  Done := False;
  repeat
    ReadMeter;
    GetSpeed;
    writeln('Loading Scan Starting Position - 5.0 A');
    LoadTargetPosition(Status.StartingWaveLength-5.0, StepsPerAngstrom);
```

```
    SetSpeed(Speed);
    writeln('Moving Grating');
    MoveGrating(Terminate);
    if (terminate) then begin
      writeln('Terminate = True @ MoveGrating ????');
    end;
    IsPortBusy(MDR, Terminate);
    Writeln('Tensioning Cable');
    TensionCable;
    ReadMeter;
    writeln('Loading Actual Scan Starting Position');
    LoadTargetPosition(Status.StartingWaveLength, StepsPerAngstrom);
    SetSpeed(10);
    writeln('Moving Grating');
    MoveGrating(Terminate);
    if (terminate) then begin
      writeln('Terminate = True @ MoveGrating ????');
    end;
    IsPortBusy(MDR, Terminate);
    ReadAbsolutePosition(WaveLength, StepsPerAngstrom);
    Writeln('Absolute Position = ',wavelength:8:3,' A');
    Writeln('Meter Reading     = ',wavelength*GratingGrooveDensity:8:3,' A');
    write('Is This Correct ? ');
    readln(resp);
    if (resp='y') or (resp='Y') then done := true;
  until done;
end; { Procedure }


Procedure GetVolt;
begin
  Write('Enter Voltage : ');
  readln(Status.PMTVoltage);
  SetVoltage(Status.PMTVoltage, Terminate);
end;


Procedure GetKeitAvg;
begin
  Write('Enter Number of Points to Average on Keithley : ');
  readln(Status.KeitAvg);
end;


Procedure GetSlit;
begin
  write('Enter InputSlit  size in Micrometers: ');
  readln(Status.InputSlit);
  write('Enter OutputSlit size in Micrometers: ');
  readln(Status.OutPutSlit);
end;


Procedure ControlTHR640(Var Terminate:Boolean);
```

```
Var Time : Array[0..2] of Real;
   Cha,I,j  : Integer;
   Inten : Double;
   Sum  : Double;
   Temp : Real;

Begin
  Terminate := False;

  { When we arrive at this point, the WaveLength of the
   Monochromator is already set to the desired value.
   It is either the Wl to be monitored or the starting WL
   of a scan if this is the first pass through.
   So we can take a reading without any change in MC settings
   until after the reading is taken.
  }

  if (Status.THR640PMT.OptEmission) then begin
    { Are We Doing Optical Emission - Watch 1 Line in Time }

    { Get the Time before and after the reading and avg to get
     the time we assign the reading to }

     MyGetTime(Time[1]);

     { CHANGED to 1 on 11/3/93 TJD
          Inten := InputKeithley(3) * -1E9; }

     if (Status.THR640PMT.Detector = 0) then
        Inten := InputKeithley(1) * -1E9
     else begin
        sum := 0.0;
        j := 0;
        for j := 1 to 10 do
              sum := sum + AtoD(1,Status.THR640PMT.Gain);
        Inten := sum/10.0;
     end;

     MyGetTime(Time[2]);
     Status.THR640PMT.Ctr := Status.THR640PMT.CTR + 1;
     if (Status.THR640PMT.Ctr > 1024) then Status.THR640PMT.Ctr := 1;

     Time[0] := (Time[1] + Time[2])/2.0 - Time0;
     if (status.TWOOPT) then
       Writeln(Opt2File, Time[0]:10:2,'  ',Inten:13:-5)
     else
           Writeln(OptFile, Time[0]:10:2,'  ',Inten:13:-5);

      { Put in DARSSParam to Save Memory }
      DARSSParam.Wave[Status.THR640PMT.CTR]  := Time[0];
      DARSSParam.SData[Status.THR640PMT.CTR] := Inten;

      { Are We Off Scale ??? }
      if (Inten > Status.THR640PMT.MAX) then begin
```

```
    View1;
    ClearViewPort;
    Status.THR640PMT.MAX := Inten * 1.1;
    SetUp1( Status.THR640PMT.XMin, Status.THR640PMT.XMax, 20.0,
        0.0, Status.THR640PMT.MAX, Status.THR640PMT.MAX/10.0, 1);
    For I := 1 to Status.THR640PMT.Ctr -1 do
     PutPixel( View1X(DarssParam.Wave[i]),
              View1Y(DarssParam.SData[i]), White);
  end;

  If (Time[0] > Status.THR640PMT.Xmax) then begin
    Temp := Status.THR640PMT.XMin;
    Status.THR640PMT.XMin := Status.THR640PMT.XMax;
    Status.THR640PMT.XMax := Status.THR640PMT.XMax +
      (Status.THR640PMT.XMax - Temp);
    DARSSParam.Wave[1]  := DARSSParam.Wave[Status.THR640PMT.CTR];
    DARSSParam.SData[1] := DARSSParam.SData[Status.THR640PMT.CTR];
    Status.THR640PMT.CTR := 1;
    View1;
    ClearViewPort;
    SetUp1( Status.THR640PMT.XMin, Status.THR640PMT.XMax, 20,
        0.0, Status.THR640PMT.MAX, Status.THR640PMT.MAX/10.0, 1);
    For I := 1 to Status.THR640PMT.Ctr -1 do
     PutPixel( View1X(DarssParam.Wave[i]),
              View1Y(DarssParam.SData[i]), White);
  end;

  View1;
  PutPixel( View1X(Time[0]), View1Y(Inten), White);

 if Keypressed then begin
   GetFuncKey(Cha);
   if (Cha = TermKey) then
     Terminate := True;
 end

end
else
 { We're Doing Wave Scan }
 begin
  { Average How Ever Many Points We Want to! }


  if (Status.THR640PMT.Detector = 0) then
    Inten := InputKeithley(Status.KeitAvg) * -1e9
  else begin
    sum := 0.0;
    j := 0;
    for j := 1 to 10 do
         sum := sum + AtoD(1,Status.THR640PMT.Gain);
    Inten := sum/10.0;
  end;

  Status.THR640PMT.Ctr := Status.THR640PMT.CTR + 1;
```

```
    if (Status.THR640PMT.Ctr > 1024) then Status.THR640PMT.Ctr := 1;

    if (status.TWOOPT) then
      Writeln(Opt2File, WaveLength:8:2,'  ',Inten:13:-5)
    else
      Writeln(OptFile, WaveLength:8:2,'  ',Inten:13:-5);

    { Put in DARSSParam to Save Memory }
    DARSSParam.Wave[Status.THR640PMT.CTR]  := WaveLength;
    DARSSParam.SData[Status.THR640PMT.CTR] := Inten;

    { Are We Off Scale ??? }
    if (Inten > Status.THR640PMT.MAX) then begin
      View1;
      ClearViewPort;
      Status.THR640PMT.MAX := Inten * 1.1;
      SetUp1( Status.StartingWaveLength, Status.EndingWaveLength,
          (Status.EndingWaveLength-Status.StartingWaveLength)/10.0,
          0.0, Status.THR640PMT.MAX, Status.THR640PMT.MAX/10.0, 0 );
      For I := 1 to Status.THR640PMT.Ctr -1 do
        PutPixel( View1X(DarssParam.Wave[i]),
                  View1Y(DarssParam.SData[i]), White);
    end;

    View1;
    PutPixel( View1X(WaveLength), View1Y(Inten), White);

    LoadAbsolutePosition(WaveLength, StepsPerAngstrom);
    SetSpeed(ScanSpeed);

    WaveLength := WaveLength + Status.WaveLengthStep;
    LoadTargetPosition(WaveLength, StepsPerAngstrom);

    MoveGrating(Terminate);
    IsPortBusy(MDR, Terminate);

    { Are We Done With The Scan ? }
    if (WaveLength > Status.EndingWaveLength) then
      Terminate := True;

    if Keypressed then begin
      GetFuncKey(Cha);
      if (Cha = TermKey) then
        Terminate := True;
    end;
  end;
End; {Procedure}


BEGIN
END.  {Unit}
```

```
UNIT Luxt;

INTERFACE
uses DOS, Crt, Misc, vars, graph, DT2811tl, MyGraph, Lifer;

CONST

  EOICode=' EOI';             {End or Identifiy Terminator Code}
  BufferedOutput='BUFFERED';     {IOTech 488 Buffered Data Transfer Code}
                          {Number Characters}
VAR
  Response          : Str255;           {IOTech Response}

PROCEDURE CheckLuxtronStatus(Var RSV,ERR,RDY,MC : Boolean; Var Mode:Integer);
PROCEDURE CloseLuxtron;
procedure InputLuxtron(Var Temp,Temp2:Real; Var LuxStatus:Boolean);
Procedure StartLuxtronInput;
{ PROCEDURE InitCommun; }
PROCEDURE InitLuxtron;
Function  IsLuxtronReady:Boolean;
Procedure ControlLuxtron;
procedure ACQ_TEMP;

IMPLEMENTATION

PROCEDURE CloseLuxtron;

begin
 {Terminate comunication via IOTech card}
  Writeln (IeeeOut, 'SEND CMD 63,55,116,55,100');
  Writeln (IeeeOut, 'CLEAR 2313');
  Writeln (IeeeOut, 'LOCAL 2313');
  { Writeln (IeeeOut, 'RESET'); }
end;

PROCEDURE CheckLuxtronStatus(Var RSV,ERR,RDY,MC : Boolean; Var Mode:Integer);
Var LuxStat,Code,tmp : Integer;
begin
  Rsv := False; ERR := False; RDY := False; MC := False; Mode := 0;
  Writeln(IeeeOut, 'SPOLL 2313');
  Readln (IeeeIn,  Response);
  Response := RemoveSpaces(Response);
  Val(Response, LuxStat, Code);
  { writeln('Luxtron Status = ',response,'  ',luxstat:4); }

  if ((LuxStat - 64) >= 0) then begin
    LuxStat := LuxStat - 64;
    RSV := True;
  end;

  if ((LuxStat - 32) >= 0) then begin
    LuxStat := LuxStat - 32;
    ERR := True;
  end;
```

```
  if ((LuxStat - 16) >= 0) then begin
    LuxStat := LuxStat - 16;
    RDY := True;
  end;

  Mode := (LuxStat and 6);

  LuxStat := Luxstat and 1;
  if (Luxstat = 1) then MC := True;

{ writeln('  RSV = ',rsv,'  ERR = ',err,'  RDY = ',rdy,'  MC  = ',MC,'  Mode = ',mode:2);
 writeln; }

end;

PROCEDURE InitLuxtron;
{Initialize IEEE communication to the Luxtron Probe}

Begin
  Writeln (IeeeOut, 'SEND CMD 63,55,101,55,114');
end;

Function IsLuxtronReady : boolean;
Var RSV,ERR,RDY,MC : Boolean;
   Mode:Integer;
begin
 CheckLuxtronStatus(RSV,ERR,RDY,MC,Mode);
 IsLuxtronReady := MC;
end;

Procedure StartLuxtronInput;
begin
  writeln(IeeeOut, 'SEND CMD 55,105')
end;


procedure InputLuxtron(Var Temp,Temp2:Real; Var LuxStatus:Boolean);
{Read data from the PMT via the Keithley 485: averaged over time}
VAR code  : Integer;
   tempresp : Str255;

begin
  {Check if data from The Luxtron is ready}
  LuxStatus := IsLuxtronReady;

  if (LuxStatus = True) then begin
    writeln(IeeeOut, 'SEND CMD 87,113');
    writeln(IeeeOut, 'ENTER 2313');
    readln(IeeeIn, response);
    Response := RemoveSpaces (Response);
    delete(response,1,6);
    if (length(response) <= 7) then begin
      Val (Response, Temp, code);
      Temp2 := 0.0;
```

```
      end
    else if (length(response) <= 18) then begin
      tempresp := copy(response,1,7);
      delete(response,1,11);
      val(tempresp, Temp, code);
      val(response, Temp2, code)
    end;

  end;

end;

Procedure ControlLuxtron;
Var LuxTime      : real;
   temp,
   temp2         : real;
   tt,tt2        : real;
begin
        if not(LuxStart) then begin
       MyGetTime(LuxTime1);
       StartLuxtronInput;
       LuxStart := True
       end
     else begin
      InputLuxtron(Temp,Temp2,LuxStatus);
          if (LuxStatus) then begin
        MyGetTime(LuxTime2);
        LuxTime := (LuxTime1 + LuxTime2)/2.0 - Time0;
        writeln(TempFile,LuxTime:8:2,temp:12:2,temp2:12:2);

        if (status.lifer = false) then begin
            if (LuxTime) > GR2XMax then begin
                           temp    := GR2XMin;
                           GR2XMin := Gr2XMax;
                           Gr2XMax := GR2XMax + (GR2XMax - temp);
                           SetUp2(GR2XMin, GR2XMax, GR2dX, GR2YMin, GR2YMax, GR2dY);
                    end;
            tt := temp;
            tt2 := temp2;
          end
        else begin
          tt := temp/20.0;
          tt2 := temp2/20.0;
        end;
        View2;

        if (tt >= Gr2YMin) and (tt <= Gr2YMax) then begin
          PutPixel(View2X(LuxTime), View2Y(tt), White);
          if (temp2 <> 0.0) then begin
            if (tt2 >= Gr2YMin) and (tt2 <= Gr2YMax) then
              PutPixel(View2X(LuxTime), View2Y(tt2), Yellow);
          end;
        end;
```

```
        LuxStart := False;
      end;
    end;
end;

procedure ACQ_TEMP;
begin
  GR2XMin :=  0.0;
  GR2XMax := 120.0;
  GR2dX   := 20.0;
  GR2YMin := -75.0;
  GR2YMax := 100.0;
  GR2dY   :=  25.0;
  SetUp2(GR2XMin, GR2XMax, GR2dX, GR2YMin, GR2YMax, GR2dY);
end;

begin end.
```

```
Unit DarssOE;

{ Code to COntrol the 1024 Element, 25 um Spaced Linear Diode Array on the
 Jarrel-Ash Monochromator with DARSS Interface }

Interface

uses crt, DT2811TL, VARS, Graph, MyGraph;


const
   Base          : Word = $228;  {These are the addresses for the DT2817.}
   Port0         : Word = $229;  {Look up DT2817 manual for deatils.}
   Port1         : Word = $22A;
   Port2         : Word = $22B;
   Port3         : Word = $22C;
   DTin0         : Byte = $E;
   DTin1         : Byte = $D;
   DTin2         : Byte = $B;
   DTin3         : Byte = $7;
   DTout0        : Byte = $1;
   DTout1        : Byte = $2;
   DTout2        : Byte = $4;
   DTout3        : Byte = $8;
   X0            : Byte = $00; {These X's are Darss Register addresses.}
   X1            : Byte = $01; {Look up the Darss Interface manual for}
   X2            : Byte = $02; {description.}
   X3            : Byte = $03;
   X4            : Byte = $04;
   X5            : Byte = $05;
   X6            : Byte = $06;
   X7            : Byte = $07;

   F10  : LongInt = 68;    { Key Code for F10 }


{*****************************************************************************}


{ Procedure StartDarss(ExposureTime : Word); }
Procedure ControlDARSS;
Procedure StartDarss(ExposureTime:Word);
Function  IsDARSSReady : Boolean;
Procedure ReadDataFromDarss(NumberOfPoints:Word; Var DataArray:ScanArray);
Procedure InitDarss;
Procedure FinishDarss;
Procedure CalcWave;
Procedure InitCalib;

Implementation


Procedure ControlDARSS;
```

```
{ Control the DARSS for LDA Acquisition : TJD

  1) Trigger Scan
  2) See if Scan is Done
  3) Read Data in PieceWise and Write Out
  4) Plot It PieceWise

}

Var
  Cont   : Boolean;
  I,J,K  : Integer;
  LDASTART,
  LDAEND        : Integer;
  TempArray     : ScanArray;
  Temp  : Word;

Begin

  Cont := True;
  LDAStart := 4;
  LDAEnd := 950;
  { If its not going, get it going! }
  If (not(DARSSParam.Started) and (Cont)) then begin
    InitDarss;
    MyGetTime(DARSSParam.Time1);
    StartDARSS(DARSSParam.Exposure);
    DARSSParam.Started := True;
    DARSSParam.Ready := False;
    DARSSParam.NumberRead := 0;
    DARSSParam.DataRead := False;
    DARSSParam.PlotTimes := 0;
    DARSSParam.DataMax := -32768;
    DARSSParam.DataMin := 32767;
    Cont := False;
  end;

  { If its going and we didnt just start it, then see if its ready }
  if ((DARSSParam.Started) and not(DARSSParam.Ready) and (Cont)) then begin
    DARSSParam.Ready := IsDARSSReady;
    Cont := True;
  end;

  { If Its Ready, And data isn't read then read it in }
  if ((DARSSParam.Ready) and not(DARSSParam.DataRead) and (Cont)) then begin

    { First Time Through, Write Header info on Scan }
    if (DARSSParam.NumberRead = 0) then begin
      MyGetTime(DARSSParam.Time2);
      Writeln(OptFile, '*****');
      Writeln(OptFile, 'DARSS Acquire from ', (DARSSParam.Time1-Time0):10:4,
      ' to ',(DARSSParam.Time2-Time0):10:4);
          Writeln(OptFile,'Average Time of Acquire is',
          ((DARSSParam.Time2+DARSSParam.Time1)/2-Time0):10:4);
```

```
    Writeln(InfoFile,'DARSS Scan At ',
    ((DARSSParam.Time2+DARSSParam.Time1)/2-Time0):10:4);
  end;

  { Read in the next DARSSParam.PointstoRead Points }
  ReadDataFromDARSS(DARSSParam.PointsToRead, TempArray);
  I := 1;
  While (I <= DARSSParam.PointsToRead) do begin
    J := I + DARSSParam.NumberRead;
    Write(OptFile, j:4,' ');
    for k := 0 to 7 do begin
      DARSSParam.Data[j+k] := TempArray[i+k];
      Write(OptFile,TempArray[i+k]:5,' ');
    end;
    I := I + 8;
    Writeln(OptFile)
  end; { While }



  DARSSParam.NumberRead := DARSSParam.NumberRead + DARSSParam.PointsToRead;

  { Have We Read in all The Data ??? }
  if (DARSSParam.NumberRead = 1024) then begin
    DARSSParam.DataRead := True;
    Cont := False;
  end;
end;

{ Now If We're Ready, Plot Out a Piece of it }
if ((DARSSParam.DataRead) and (Cont)) then begin
  Inc(DARSSParam.PlotTimes);
  Case DARSSParam.PlotTimes of
    1 : Begin
      { Don't use 1024 as Diodes 1023 & 1024 are screwed up }
      { Assign Data Max and Min Values }
      For I := LDASTART to LDAEND do begin

        Temp := DARSSParam.Data[i] - DARSSParam.BData[i];

        if (Temp > 32767) then
          { DARSSParam.PData[i] := Temp - 65536}
           DARSSParam.Pdata[i] := 0
        else
          DARSSParam.PData[i] := Temp;

        if (DARSSParam.PData[i] > DARSSParam.DataMax) then
          DARSSParam.DataMax := DARSSParam.PData[i];
        if (DARSSParam.PData[i] < DARSSParam.DataMin) then
          DARSSParam.DataMin := DARSSParam.PData[i];
      end;
      end;
      { Scale to 0 - 1 for Plotting }
    2 : Begin
```

```
        With DARSSParam do
          SetUp1(Wave[LDAStart], Wave[LDAEnd],
                 (Wave[LDAEnd]-Wave[LDASTart]),
               -200, DataMax, (DataMax+200), 0 );
        end;
      3 : Begin
        View1;
        For I := LDASTart to 511 do
          PutPixel(view1X(DARSSParam.Wave[i]),
                   view1Y(DARSSParam.PData[i]*1.0), White);
        end;
      4 : Begin
        View1;
        For I := 512 to LDAEnd do
          PutPixel(view1X(DARSSParam.Wave[i]),
                   view1Y(DARSSParam.PData[i]*1.0), White);
        DARSSPAram.Started := False;
        end;



    end; { Case }
  end; { If }
End; { Procedure ControlDARSS}


Function Bnot(i: Word): Byte;
{ Return the LSB Compliment of a 2 Byte Word by Trashing MSB }

var b: word; b1: byte;

Begin
  B    := not(i);
  B1   := Trunc(b/256);     { This MSB Is disregarded. }
  Bnot := b - b1*256;
End;


procedure DToutput(Address : Word; Data   : Byte);
{This procedure puts out the value 'Data' to the corresponding DT port}

Begin

{This Case statement sets only the bit which is required for the
particular Port.  It leaves all the other bits as they were}

   Case Address of
    $229 : If (($01 and Port[Base]) = 0) then
                   Port[Base] := Port[Base] + DTout0;
    $22A : If (($02 and Port[Base]) = 0) then
                   Port[Base] := Port[Base] + DTout1;
    $22B : If (($04 and Port[Base]) = 0) then
                   Port[Base] := Port[Base] + DTout2;
    $22C : If (($08 and Port[Base]) = 0) then
```

```
                  Port[Base] := Port[Base] + DTout3;
   End; {Case}
   Port[Address] := Data; {Send data to the port}
 End; {DToutput}


procedure DTinput(Address : Word; var Data: Byte);
{This procedure reads 'Data' from the selected port}

Begin {DTinput}

{Again, this Case statement makes sure that only the bit required for
the particular Port is set to 0.  It leaves all the other bits alone}

  Case Address of
    $229 : If ((Port[Base] and $01) = 1) then
                      Port[Base] := Port[Base] and DTin0;
    $22A : If ((Port[Base] and $02) = 2) then
                      Port[Base] := Port[Base] and DTin1;
    $22B : If ((Port[Base] and $04) = 4) then
                      Port[Base] := Port[Base] and DTin2;
    $22C : if ((Port[Base] and $08) = 8) then
                      Port[Base] := Port[Base] and DTin3;
  End; {Case}

  Data := Port[Address];
End; {DTinput}


procedure ResetDarssPorts;
{This procedure resets the Darss Ports.}

Begin {ResetDarssPorts}
   DToutput(Port1, $DF); {DENA=1;INIT_L=0;AQR_ENA_L=1;CLR_L=1}
   DToutput(Port0, $FF);
   DToutput(Port2, Bnot($00)); {All 0's}
   DToutput(Port3, Bnot($00)); {All 0's}
End; {ResetDarssPorts}


procedure SetDarssRegister(Register: Byte; LSB: Byte; MSB: Byte);
{This Procedure sends the Least Significant Byte (LSB) and MSB
 to the chosen Darss register}

Begin {SetDarssRegister}

  DToutput(Port3, Bnot(Register));
  DToutput(Port2, Bnot(LSB));
  DToutput(Port3, Bnot(Register + $20)); {Write Strobe}
  If ((Register <> X3) and (Register <> X7)) then
                   {i.e. if it's a two byte register}
    Begin
      DToutput(Port3, Bnot(Register));
      DToutput(Port2, Bnot(MSB));
```

```
    DToutput(Port3, Bnot(Register + $20));
   End; {If}
  DToutput(Port3, Bnot(Register));
End; {SetDarssRegister}


function ReadDarssRegister(Register : Byte): Byte;
{This function reads the particular Darss Register and returns the value}

var InData : Byte;

Begin {ReadDarssRegister}
    DToutput(Port2, Bnot($00)); {Reset Port2}
    DToutput(Port3, Bnot(Register)); {Select the Register}
    DToutput(Port3, Bnot(Register + $40)); {The Read Strobe}
    DTinput(Port2, InData);
    ReadDarssRegister := Bnot(InData);
    DToutput(Port3, Bnot(Register)); {Disable read/write strobes}
    DToutput(Port2, Bnot($00)); {Re-initialize Port2}
End; {ReadDarssRegister}


procedure SetDarssControl(ExposureTime : Word);
{This procedure sets the scan parameters for the Darss interface.
Note that all values assigned to various Darss registers (X0-X7)
are complements of the values assigned in the Main Etcher program.
This is because the Data bus for Darss is low-level active, while
DT2817 is not. The ProLog STD 7000 board on main etcher is low
level active}

const Exponent      : Byte = $02;
    NumScans       : Byte = $02;

var  UpperTimeByte   : Byte;
     LowerTimeByte   : Byte;

Begin {SetDarssControl}

  DToutput(Port1, Bnot($00));

  { Set Continuous normal mode and 2^14 resolution }
  SetDarssRegister(X0, $01, $00);

  { Set Source ADC to 0 and time Exponent }
  SetDarssRegister(X2, $00, Exponent);

  { Compute bytes for Integration Time (from ExposureTime). These
  will be used to set the integration time using X4.}
  UpperTimeByte := (ExposureTime-10) DIV 256;
  LowerTimeByte := (ExposureTime-10) MOD 256;

  { Set Configuration and Counters.
  First, all the addresses for selecting counters are sent to X7.
  Then the necessary values are sent to the counters (registers
```

```
  X4-X6). The first SetDarssRegister command sends values for
  counters X4 and X5, to X7. The next set of 3 DToutput commands
  are just truncated SetDarssregister commands, and send value for
  counter X6 to X7. }

  SetDarssRegister(X7, $34, $74);
  DToutput(Port2, Bnot($B4));
  DToutput(Port3, Bnot(X7 + $20));
  DToutput(Port3, Bnot(X7));

  {Now the counters have been selected, load X4}
  SetDarssRegister(X4, LowerTimeByte, UpperTimeByte);

  {Set Breakpoint clear }
  SetDarssRegister(X1, $00, $44);

  { X3 and X6 don't need to be set for Continuous normal mode }
  { This is for number of scans }
  SetDarssRegister(X5, NumScans, $00);

End; {SetDarssControl}


procedure StartDarss(ExposureTime : Word);
{This procedure uses SetDarssControl to set the Darss, and then sends
the AQR_ENA_L signal to start the acquisition}

Begin
  ResetDarssPorts;
  SetDarssControl(ExposureTime);

  {Reset the Darss data Ports}
  DToutput(Port0, Bnot($00));
  DToutput(Port3, Bnot($00));
  DToutput(Port2, Bnot($00));

  {Set AQR_ENA_L to 0 - Tell Darss to aqr data}
  DToutput(Port1, Bnot($10));

end; {Start_Darss}


Function IsDARSSReady : Boolean;
begin
{ Examine StoreL to see if its done with scan yet... }
  DTinput(Port0, StoreCheck);
  if ((StoreCheck and $10) = 0) then
    IsDARSSReady := True
  Else
    IsDARSSReady := False;

end;
```

```
procedure ReadDataFromDarss(NumberOfpoints:Word; Var DataArray:ScanArray);
var i: Word;
        ch : char; endit : boolean;

Begin
  { Start Data transfer. Basically what's happening in the Repeat
   loop is that we're trying to read the 3rd bit of X0. The
   3rd bit of X0 is the 'Acquiring Status' bit. This bit goes high
   once the AQR_ENA_L is set to 0.

   Send the AQR_ENA_L signal again, and also set DENA to 1}

  DToutput(Port1, Bnot($10));

  for I := 1 to NumberOfPoints do Begin
    endit := false;
    Repeat
      DTinput(Port0, StoreCheck);
      if keypressed then begin
        ch := readkey;
        if ord(ch) = 0 then begin
          ch := readkey;
          if ord(ch) = F10 then endit := true;
        end;
      end;
      { Wait until the DARSS is ready or the user exits }
    Until ((StoreCheck and $10) = 0) or endit;
    DTinput(Port2, LoByte);
    DTinput(Port3, HiByte);
    DataArray[i] := Bnot(LoByte) + Bnot(HiByte)*256;
    DToutput(Port1, Bnot($11));
    { Signal DCI that value accepted
     using CLR (low-level) flag, & Keep AQR_ENA asserted }
    { CHANGED FROM $01, $00 to $11, $10 to keep <AQR_ENA> Enabled
     TJD 5/17/91 }
    DToutput(Port1, Bnot($10));
  End;

End; {ReadDataFromDarss}


procedure InitDarss;
begin ResetDarssPorts; end;


procedure FinishDarss;
begin ResetDarssPorts; end;


Procedure CalcWave;
var
  cal,   cal1, cal2, r : Real;
  I : Integer;
```

```
begin
  With DARSSParam do begin
    Cal  := DARSSCalib.Cal[ Grating ];
    Cal1 := DARSSCalib.Cal1[ Grating ];
    Cal2 := DARSSCalib.Cal2[ Grating ];
    R    := DARSSCalib.R[ Grating ];
    if (Grating <> 2) then
        Cal1 := DARSSCalib.a11 - DARSSCalib.a22*Center
                 - DARSSCalib.a33*Center*Center
    else
      Cal1 := Cal1 + Cal2*Center;
    For I := 1 to 1024 do
      Wave[i] := (Center + (i-512)/1023.*Cal1)*R
                 -Cal + (i-512)*(i-512)*Cal2 + Correction;
  end; { With DARSSParam }
end; { Procedure }

Procedure InitCalib;
Begin
  With DARSSCalib do begin
    Cal[1]  := -1.17E2;
    Cal1[1] := 6.775e2;
    Cal2[1] := 0.0;
    Cal[2]  := -6.717E2;
    Cal1[2] := 7.543E2;
    Cal2[2] := -8.002E-5;
    Cal[3]  := -4.4e1;
    Cal1[3] := 9.5e2;
    Cal2[3] := 0.0;
    R[1]    := 1;
    R[2]    := 8;
    R[3]    := 1;
    A11     := 9.5E2;
    A22     := 5.88E-2;
    A33     := -2.1E-6;
  end; { DARSSCalib }
end;

begin
End. {Unit}
```

```
Unit Comdel;

INTERFACE
Uses Dos, Crt, AuxInOut, Misc, DT2811TL, VARS, v2;


TYPE

  ComdelRec = Record
    Power       : Real;
    Voltage     : Real;
    Current     : Real;
    Impedance   : Real;
    Phase       : Real;
    DCBias      : Real;
  end;


VAR
  Com2In,
  Com2Out       : Text;                 { Com2 Files for Comdel }
  ComdelData    : ComdelRec;            { Comdel RPM Info }
  Ch            : Char;


Procedure InitCom2Port;
Procedure CloseCom2Port;
Procedure SetUpComdel;
Procedure GetComdelPoint;
Procedure ClearCom2;


IMPLEMENTATION


{ ************************************************************************ }
Procedure InitCom2Port;
Begin
{
  $C3 = 11000011
  Baud Rate  = 4800 = 110
  Parity     = None =  X0
  StopBit    = 1    =  0
  WordLength = 8    =  11

  From IBM Tech Ref. Manual, Pg. 5-43, Interupt 14, RS232_IO

  Also, Even though the variables are labeled COM2, use COM1
  Port by making the second arguement in the assignaux statements
  a 0.  a 0 -> COM1, a 1 -> COM2
}
  AssignAux (Com2In,0,$C3);
  AssignAux (Com2Out,0,$C3);
  ReWrite (Com2In);
```

```
  ReSet (Com2Out);
end;


{ ************************************************************************ }
Procedure CloseCom2Port;
Begin
  Close(Com2In);
  Close(Com2Out);
end;


{ ************************************************************************ }
Procedure ClearCom2;
Var J : integer; Ch : Char;

begin
  J := 1;
  Repeat
    Read(Com2Out, ch);

{    if (ch<>#26) then writeln(ivfile,'CLEARCOM2: ',ch); }

    J := J + 1;
  Until (J = 200) or (Ch = '>');
end;


{ ************************************************************************ }
Procedure SetUpComdel;
{ Set Comdel to DISABLE ECHO of Characters sent to it! }
begin
  write(Com2In, 'DE'+chr(13));
  ClearCom2;
end;


{ ************************************************************************ }
{ See if the Data Read in is Valid and if so, Parse it. }
Procedure IsComdelDataValid(s: String; Var ValidData: Boolean);
var i,j,k,l,NumSpace,NumChar,Num,Code : integer; S2 : String;
begin
  l := ord(s[0]);
  i := 3;
  ValidData := False;
  NumSpace := 0;
  NumChar := 0;
  S2 := '';
  if ( (s[1]=#10) and (s[2]=#13) and (s[l-2]=#10) and
      (s[l-1]=#13) and (s[l] = '>') ) then begin
       for j := 3 to l-3 do begin
          Case s[j] of
            #32                  : begin
               NumSpace := NumSpace + 1;
```

```
          { A Space Means we have finished a Number, therefor,
            We can move the present number to another string
            and convert it into a number and place it in the
            correct variable }
          For k := i to j-1 do
            S2[k-i+1] := s[k];
          S2[0] := chr(j-i);
          i := j + 1; {Set Pointer to the start of the next number}
          {$R-}
          Val(S2, num, code);
          {$R+}
          if (Code <> 0) then exit;
          Case NumSpace of
           1 : ComdelData.Power     := Num;
           2 : ComdelData.Voltage   := Num;
           3 : ComdelData.Current   := Num / 10.0;
           4 : ComdelData.Impedance := Num / 10.0;
           5 : ComdelData.Phase     := Num / 10.0;
           6 : ComdelData.DCBias    := Num;
          else
            exit;
          end; {Case NumSpace}
        end; {Begin #32}
        '0'..'9', '-'          : NumChar := NumChar + 1;
          else
        exit;
      end; {Case S[j]}
    end { for }
  end { if () and () and () and () and () }
 else
   exit;
 if (NumSpace = 6) then ValidData := True;
end;


{ ************************************************************************ }
Procedure GetComdelPoint;
Var T1, T2 : Real; S,S1 : String; I,J,K : Integer; CH,CH2 : Char;
   ValidData : Boolean;
begin

  ValidData := False;

  REPEAT { Until We Read in a Valid Data String! }

    S[0] := Chr(0);

    { First, Delete things in the buffer that Should Not Be There }
    write(com2in, chr(8)); write(com2in, chr(8));
    write(com2in, chr(8)); write(com2in, chr(8));
    write(com2in, chr(8)); write(com2in, chr(8));
    write(com2in, chr(8)); write(com2in, chr(8));
    write(com2in, chr(8)); write(com2in, chr(8));
    write(com2in, chr(8)); write(com2in, chr(8));
```

```
write(com2in, chr(8)); write(com2in, chr(8));
write(com2in, chr(8)); write(com2in, chr(8));
write(com2in, chr(8)); write(com2in, chr(8));
write(com2in, chr(8)); write(com2in, chr(8));

{ Now send the MD Command, using the HandShaking Version,
  HD, to read the RPM Status }

write(com2in, 'H');
write(com2in, 'D'); write(com2in, chr(13));

MyGetTime(T1); {Now that the Command has been Sent, Mark the Time }

J := 1; I := 1;
Repeat { Read in a Response }
  { If works by Handshaking.  Sending a charecter to us each
    time we send a character to it.  We sent "HD<CR>"  On the
    first time through, we do not to send a <CR> as a char is already
    waiting for us.  After that, send a <CR> in order to receive a
    character from the Comdel }

  if (i <> 1) then
    write(Com2In, CHR(13));

      Read(Com2Out, Ch);
      I := I + 1; { Count Total Number of Reads }
  { Is it a ^Z ??? If So, Skip it! }
      if (ch <> #26) then begin
    { Add a Valid Character to the String and Adjust the
      String Length }
    S[J] := Ch;
    S[0] := Chr(j);
    J := J + 1;
      end;

  { Look for Key Board Activity, under "SPECIAL" keys }
      ch2 := #0;
{     if (keypressed) then begin
    ch2 := readkey;
    if (ch2 = #0) then
      ch2 := readkey
    else
      ch2 := #0;
  end;
}
  { Is the Last Character a valid ">", but the total length
    less than or equal to 7 ?  This is too short to be a valid
    answer, but may insted be just a command line prompt.
    In this case, reset the counters and begin
    reading again after issuing a new read command. }
  if ( (ch='>') and (J <= 7)) then begin
    I := 1;
    J := 1;
        ch := '*';
```

```
        write(com2in, chr(8)); write(com2in, chr(8));
            write(com2in, chr(8)); write(com2in, chr(8));
            write(com2in, chr(8)); write(com2in, chr(8));
            write(com2in, chr(8)); write(com2in, chr(8));
            write(com2in, chr(8)); write(com2in, chr(8));
            write(com2in, chr(8)); write(com2in, chr(8));
            write(com2in, chr(8)); write(com2in, chr(8));
            write(com2in, chr(8)); write(com2in, chr(8));
            write(com2in, chr(8)); write(com2in, chr(8));
            write(com2in, chr(8)); write(com2in, chr(8));
        write(com2in, 'H');
        write(com2in, 'D'); write(com2in, chr(13));
      end;
      { Exit if, 1) We reach a valid end of read,
        2) we read too many "good" characters,
            3) Someone presses F10 or
            4) We had too many total reads (good + ^Z)  }
    until (ch='>') or (j=250) or (ch2 = #68) or (l = 500);

    if (l <> 500) then begin
        S[0] := chr(j-1);
        MyGetTime(T2); { Take another time reading at the end of the read }
      S1 := S;
        IsComdelDataValid(s, ValidData);
      S := S1;
    End;

  Until (ValidData = True);

{  writeln(IVFile);
   Writeln(IVFile,((T1+T2)/2-Time0):10:4,'  ',s);
   write(ivfile, s[3],'  '); }

  With ComdelData Do
  Writeln(IVFile, (((T1+T2)/2)-Time0):10:4, '  ',
    Power:5:0,' ',Voltage:5:0,' ',Current:6:1,' ',
    Impedance:6:1,' ',Phase:6:1,' ',DCBias:5:0);

end; {Proc}

BEGIN
END.
```

```
Unit Vars;
Interface

TYPE
  str1   = string[1];
  str7   = string[7];
  str8   = string[8];
  str12  = string[12];
  str60  = string[60];
  str255 = string[255];
  StringKind = (Alfa, Numeric, Mix);

  FileStat = (Open, Closed);

  THR640PMTRec = Record
    WaveScan    : Boolean;
    OptEmission : Boolean;
    Detector        : Integer;          { Detector Type, 0 = Keithley 485
                                          1 = Keithley 427 Fast I -> V }
    Gain            : Integer;          { Detector 1 Gain Setting }
    Gain2           : Integer;          { K427 Gain, 10^Gain2 }
    RiseTime        : Integer;          { K427 Rise Time, Msec }
    CTR             : Integer;          { Array Index }
    MAX             : Real;         { Plot Y Max Value }
    XMax,XMin       : Real;         { X Max and Min Values for Plot }
  end;

  StatRec = Record
    Lifer           : Boolean;          { Laser InterFerometry ? }
    Comdel          : Boolean;          { Use Comdel ? }
    THR640PMT : THR640PMTRec;                  { THR640 PhotoMultiplier Tube ? }
    THR640LDA : Boolean;                { THR640 Linear Diode Array ? }
    DARSS     : Boolean;            { DARSS Diode Array ? }
    THR640    : Boolean;            { Using the 640 at all ? }
    LUXTRON   : Boolean;                { Use The Luxtron }
    TWOOPT    : Boolean;                { Do TWO Opticals at once ??? }
    THR640MirrorPos : Char;             { Mirror Position : A or L }
    THR640Grating   : Char;             { Grating: A or B }
    FileBase  : STR8;                   { Run Sequence File Base }
    Path      : String;             { Path to Files }
    PMT             : Boolean;          { Use THR640 Lateral Exit? }
    LDA             : Boolean;          { Use THR640 Axial Exit ? }
    Grating         : Char;             { Which Grating A/B }
    RunNumber : Integer;            { Run Number in Series }
    Files           : FileStat;         { Open or Closed ? }
    StartingWaveLength : Real;      { Staring WaveLength for Scan
                                          or Fixed WaveLength }
    EndingWaveLength      : Real;  { End Value in Angstroms }
    WaveLengthStep        : Real;  { Step Value in a Scan }
    PMTVoltage   : Real;  { Cathode Voltage for PMT }
    KeitAvg               : Integer;        { # Pts to Avg on Keithley }
    InputSlit             : Real;  { 640 Input Slit in uM }
    OutputSlit     : Real;  { 640 Output Slit in uM }
  end;
```

```
ComdelRec = Record
  Power       : Real;
  Voltage     : Real;
  Current     : Real;
  Impedance   : Real;
  Phase       : Real;
  DCBias      : Real;
end;

ScanArray = Array [1..1024] of word;

DARSSRec = Record
  Grating      : Integer;        { Grating 1,2, or 3 }
  Center       : Real;           { Center Wavelength in A }
  Exposure  : Word;              { Exposure Time in ms }
  Correction: Real;              { Angstrom Correction to WaveLength }
  Started      : Boolean;        { Scan Started ??? }
  Ready        : Boolean;        { Scan Done ? }
  DataRead     : Boolean;        { Finished Reading Data ? }
  Time1        : Real;           { Scan Start Time }
  Time2        : Real;           { Scan End Time }
  NumberRead: Integer; { Number of Points Read in }
  BData        : ScanArray;      { BackGround Data }
  Data         : ScanArray;      { Data }
  PData        : Array[1..1024] of integer; { Plot Data }
  SData        : Array[1..1024] of Real; { Scaled Data }
  Wave    : Array[1..1024] of Real; { WaveLength Conversion of Cells}
  PointsToRead : Integer;        { Number to Read at 1 Time }
  DataMax      : Integer;        { Max Value of Data }
  DataMin      : Integer;        { Min Value of Data }
  PlotTimes    : Integer;        { Number of Times in Loop }
end;


CalibRec = Record
  Cal  : Array[1..3] of Real;
  Cal1 : Array[1..3] of Real;
  Cal2 : Array[1..3] of Real;
  R    : Array[1..3] of Real;
  A11, A22, A33 : Real;
end;

const
 F1   : char = #59;
 F2   : char = #60;
 F3   : char = #61;
 F4   : char = #62;
 F5   : char = #63;
 F6   : char = #64;
 F7   : char = #65;
 F8   : char = #66;
 F9   : char = #67;
 F10  : char = #68;
 Home : char = #71;
```

```
Up    : char = #72;
PgUp  : char = #73;
Left  : char = #75;
Right : char = #77;
End_  : char = #79;
Down  : char = #80;
PgDn  : char = #81;

Again : Integer = 0;
First : Integer = 1;

HexChar : Array[0..15] of Char = ('0','1','2','3','4','5','6','7', '8','9','A','B','C','D','E','F');
DblHexChar : Array[0..31] of char = ('0','1','2','3','4','5','6','7','8','9','A','B','C','D','E',
'F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V');

 MixSet : Set of Char = ['A'..'Z', '0'..'9', '-'];
 YesSet: Set of Char=['y','Y'];  {Characters Meaning Yes}
 NumSet: Set of Char=['-','0','1','2','3','4','5','6','7','8','9'];
                         {Number Characters}

 InfoPath      : String = 'INFO\';
 LiferPath     : String = 'LIFER\';
 OptPath       : String = 'OPT\';
 IVPath        : String = 'IV\';
 PLOTPath      : String = 'PROPLOT\';
 TEMPPath      : String = 'TEMPERAT\';


VAR
 FuncKey       : Integer;                { Ordinal Value of Key Pressed }
 InfoFile      : Text;                   { Run Information File }
 LiferFile     : Text;                   { Laser Interferometry File }
 OptFile       : Text;                   { Optical Emission File }
 Opt2File      : Text;                   { A Secondary OPT File }
 IVFile        : Text;                   { Comdel Output File }
 TempFile      : Text;                   { Temperture File }
 OutFile       : Text;            { General File }
 Status : StatRec;                 { Overall Status Shit }
 Time0         : Real;                   { Time Zero Point }
 ComdelData    : ComdelRec;              { Comdel RPM Info }
 OutFileString : String;
 a,b,c,ReprocName: string;

 GoodScan     : boolean;                 {Flag to check if scan is good.}
 LoByte      : Byte;             {LoByte of data.  Comes over Port2}
 HiByte      : Byte;             {HiByte of data.  Comes over Port3}
 StoreCheck   : Byte;   {Byte to check if 'Store' flag is set. Port0}
 DARSSParam    : DARSSRec;               { DARSS Info }
 DARSSCalib    : CalibRec;
 ShowLifer     : Boolean;

 FName : String;
 ExtremaTimes : Array[1..50] of real;
 ExtremaType  : Array[1..50] of String;
```

```
  NumberExtrema: integer;

  ACTYMin,
  ACTYMax,
  RAWYMin,
  RAWYMax      : Real;

  LiferTMax    : Real;  {Maximum Time}

  LuxStart     : Boolean;              { Luxtron Read Started ? }
  LuxTime1,
  LuxTime2     : Real;                 { Luxtron Times }
  RSV,ERR,RDY,
  MC,LuxStatus : Boolean;
  Mode         : Integer;
  MagSync      : Integer;
  Num_Rise     : Integer;
  Prev_Rise    : Integer;
  Mag_Field    : Integer;
  Prev_Field   : Integer;
  MagModNum    : Integer;

  IeeeOut, IeeeIn : Text;     {Files for IOTech 488 communication}

Implementation

BEGIN
END.
```

```
Unit AuxInOut;

{This Unit implements a device driver for the communication (serial
 ports) of the IBM PC.  This code has been directly lifted from the
 Turbo Pascal 4.0 manual on page 365.

 L. Baston -- Rev 0 -- 5/11/88}

Interface
Uses Dos;

Procedure AssignAux(var F:Text; Port, Params:word);

Implementation
{$R-,S-}

Type
  AuxRec = Record
    Port,Params: word;
    Unused: array[1..12] of byte;
  end;

Procedure AuxInit(Port,Params:word);
inline(
  $58/         {POP AX   ;Pop parameters}
  $5A/         {POP DX   ;Pop port number}
  $B4/$00/     {MOV AH,0 ;Code for Initialize}
  $CD/$14);    {INT 14H  ;Call BIOS}

Function AuxInChar(Port:word):Char;
inline(
  $5A/         {POP DX   ;Pop port number}
  $B4/$02/     {MOV AH,2 ;Code for input}
  $CD/$14);    {INT 14H  ;Call BIOS}

Procedure AuxOutChar(Port:word; Ch:Char);
inline(
  $58/         {POP AX   ;Pop parameters}
  $5A/         {POP DX   ;Pop port number}
  $B4/$01/     {MOV AH,1 ;Code for output}
  $CD/$14);    {INT 14H  ;Call BIOS}

Function AuxInReady(Port:word):Boolean;
inline(
  $5A/         {POP DX   ;Pop port number}
  $B4/$03/     {MOV AH,3 ;Code for status}
  $CD/$14/     {INT 14H  ;Call BIOS}
  $88/$E0/     {MOV AL,AH;Get line status in AH}
  $24/$01);    {AND AL,1 ;Isolate Data Ready bit}
{$F+}

Function AuxInput(var F:TextRec):integer;
var
  P:Word;
```

```
begin
  with F,AuxRec(UserData) do begin
    P:=0;
    while AuxInReady(Port) and (P<Bufsize) do begin
      BufPtr^[P]:=AuxInChar(Port);
      Inc(P);
    end;
    BufPos:=0;
    Bufend:=P;
  end;
  AuxInput:=0;
end;

Function AuxOutput(var F:TextRec):integer;
var
  P:Word;
begin
  with F,AuxRec(UserData) do begin
    P:=0;
    while P<BufPos do begin
      AuxOutChar(Port,BufPtr^[P]);
      Inc(P);
    end;
    BufPos:=0;
  end;
  AuxOutput:=0;
end;

Function AuxIgnore(var F:TextRec):integer;
begin
  AuxIgnore:=0;
end;

Function AuxOpen(var F:TextRec):integer;
begin
  with F,AuxRec(UserData) do begin
    AuxInit(Port,Params);
    if Mode=fmInput then begin
      InOutFunc:=@AuxInput;
      FlushFunc:=@AuxIgnore;
    end else begin
      Mode:=fmOutput;
      InOutFunc:=@AuxOutput;
      FlushFunc:=@AuxOutput;
    end;
    CloseFunc:=@AuxIgnore;
  end;
  AuxOpen:=0;
end;

{$F-}

Procedure AssignAux;
begin
```

```
  with TextRec(F) do begin
    Handle:=$FFFF;
    Mode:=fmClosed;
    BufSize:=Sizeof(Buffer);
    BufPtr:=@Buffer;
    OpenFunc:=@AuxOpen;
    AuxRec(UserData).Port:=Port;
    AuxRec(UserData).Params:=Params;
    Name[0]:=#0;
  end;
end;
end.
```

Appendix 2

AME-5000 Data Analysis and Plotting Code

This appendix contains listing of the source code used to analyze and plot data taken on the AME-5000. All of this code was written in Borland Turbo Pascal 6.0 and run on an 80386 PC. Many of the program units used here are identical to those used by **Gather** and were listed above. For this reason, only new routines are listed here. The following list outlines the program units that are listed in this section, and their function.

Msopt: Main Program 533
Msoptt: Decode .INF file. 557
Msoptg: Screen Plot of DARSS OE Spectra 561
Plot: Produce Proplot Code for Lifer, Luxtron, OE, and Actinometry 565
Plot2: Produce Proplot Code for Comdel RPM 582
MsOptVg: Global Variables 597
Lifanal: Laser Interferometry Analysis. 599

The listings for each of these units follows.

```
program msopt;

{ MASSAGE for Optical Emission Data }

uses misc, crt, dos, vars, darssoe, msoptt,
   msoptg, graph, plot, plot2, msoptvg, lifanal;

VAR
  TSEQ          : Integer;         { Temperature Plot File Sequence Number }
  LSEQ          : Integer;         { Lifer Plot File Sequence Number }
  SSEQ          : Integer;         { Spectra Plot Sequence Number }
  ASEQ          : Integer;         { Actinometry Plot Sequence Number }
  ESEQ          : Integer;         { Electrical Sequence }

  AUTO                  : Boolean;         {DO IT ALL AUTO USING YNYNYY(-1)N}
  StartRun      : Integer;
  TempStr       : String;
  f             : Text;

Procedure FileNameGen(Var Success:Boolean);
{
  Generate a Unique File Name.

  IYYMDCNN

  I  = User Initial
  YY = Year - 1900 in Base 16 (1991 - 1900 = 91 = $5B)
  M  = Month in Base 16
  D  = Day of Month in Base 36
  C  = Chamber Letter (A or B)
  NN = Run ID Number 01 -> 99

  This Subroutine examines the default path subdirectory INFO
  for .inf files until its ok.
}


Var
  Tens    : Integer;
  FileName : String;
  DirInfo  : SearchRec;     { PreDefined in DOS Unit }

begin
  Success := False;
  Status.FileBase[0] := Chr(8);
  Repeat
    Status.RunNumber := Status.RunNumber + 1;
    if (Status.RunNumber > 99) then begin
      Writeln('Run Sequence Full - Change User Initial');
      Exit;
    end;
    if (Status.RunNumber) < 10 then begin
      Status.FileBase[7] := '0';
      Status.FileBase[8] := Chr( Ord('0') + Status.RunNumber);
```

```
      end
    else begin
      Tens := Status.RunNumber div 10;
      Status.FileBase[7] := Chr( Ord('0') + Tens);
      Status.FileBase[8] := Chr( Ord('0') + Status.RunNumber - 10*Tens);
    end;
    FileName := Status.Path+InfoPath+Status.FileBase+'.INF';
    FindFirst(FileName, AnyFile, DirInfo);
  Until (DosError <> 0);
  Success := True;
end; {Procedure}


procedure CountRuns(Var NumberofRuns : Integer);

VAR
  DirInfo  : SearchRec;     { PreDefined in DOS Unit }

begin
  Interrogate;
  Status.RunNumber := 0;
{   Status.FileBase := 'T5B5RB00';
   Status.Path := 'E:\TIM\EXPERIME\';
}
   Status.FileBase[0] := Chr(8);
  Repeat
    Status.RunNumber := Status.RunNumber + 1;
    if (Status.RunNumber > 99) then begin
      Writeln('Run Sequence Full - Change User Initial');
      Exit;
    end;
    if (Status.RunNumber) < 10 then begin
      Status.FileBase[7] := '0';
      Status.FileBase[8] := Chr( Ord('0') + Status.RunNumber);
      end
    else begin
      Tens := Status.RunNumber div 10;
      Status.FileBase[7] := Chr( Ord('0') + Tens);
      Status.FileBase[8] := Chr( Ord('0') + Status.RunNumber - 10*Tens);
    end;
    FileName := Status.Path+InfoPath+Status.FileBase+'.INF';
    FindFirst(FileName, AnyFile, DirInfo);
  Until (DosError <> 0);
  NumberOfRuns := STatus.RunNumber -1;
end; {Procedure}


procedure MakeFileName(FNumber : Integer);

begin
    Status.FileBase[0] := Chr(8);
    if (FNumber) < 10 then begin
      Status.FileBase[7] := '0';
      Status.FileBase[8] := Chr( Ord('0') + FNumber);
```

```
      end
    else begin
      Tens := FNumber div 10;
      Status.FileBase[7] := Chr( Ord('0') + Tens);
      Status.FileBase[8] := Chr( Ord('0') + FNumber - 10*Tens);
    end;
end; {Procedure}


Procedure ReadInfo;
begin
  with STATUS do begin
    Lifer     := False;
    Comdel    := False;
    THR640PMT.WaveScan := False;
    THR640PMT.OptEmission := False;
    THR640LDA := False;
    DARSS     := False;
    THR640    := False;
    PMT           := False;
    LDA           := False;
    LUXTRON   := False;
  end; {With}
  readln(InfoFile, Dummy);
  Repeat
    Readln(InfoFile,Dummy);
  Until (Dummy[1] = '*') and (Dummy[0] > #1);
  with Status do begin
    Readln(InfoFile, DUMMY);
          if (DUMMY = 'TRUE') then LIFER := True Else LIFER := False;
    Readln(InfoFile, DUMMY);
      if (DUMMY = 'TRUE') then DARSS := True Else DARSS := False;
    if (DARSS) then begin
      readln(InfoFile, DARSSParam.Grating, DARSSParam.Center);
      readln(InfoFile, DARSSParam.Exposure, DARSSParam.Correction);
    end; { if DARSS }
    Readln(InfoFile, DUMMY);
      if (DUMMY = 'TRUE') then COMDEL := True Else COMDEL := False;
    Readln(InfoFile, DUMMY);
      if (DUMMY = 'TRUE') then THR640 := True Else THR640 := False;
    if (THR640) then begin
      readln(InfoFile, Grating);
      readln(infofile, inputslit);
      readln(infofile, outputslit);
      Readln(InfoFile, DUMMY);
      if (DUMMY = 'TRUE') then LDA := True Else LDA := False;
      Readln(InfoFile, DUMMY);
      if (DUMMY = 'TRUE') then PMT := True Else PMT := False;
      if (PMT) then begin
        Readln(InfoFile, PMTVoltage);

        Readln(InfoFile, THR640PMT.Detector);
        Readln(InfoFile, THR640PMT.Gain);
        Readln(InfoFile, THR640PMT.Gain2);
```

```
        Readln(InfoFile, THR640PMT.RiseTime);

        Readln(InfoFile, DUMMY);
        if (DUMMY = 'TRUE') then THR640PMT.OptEmission := True
                          Else THR640PMT.OptEmission := False;
        if (THR640PMT.OptEmission) then
          readln(InfoFile, StartingWaveLength);
        Readln(InfoFile, DUMMY);
        if (DUMMY = 'TRUE') then THR640PMT.WaveScan := True
                          Else THR640PMT.WaveScan := False;
        if (THR640PMT.WaveScan) then Begin
          readln(InfoFile, StartingWaveLength);
          readln(InfoFile, EndingWaveLength);
          readln(InfoFile, WaveLengthStep);
          readln(InfoFile, KeitAvg);
            end; { if WaveScan }
      end; { if PMT }
    end; { If THR640 }
    readln(InfoFile, DUMMY); { PATH }
    readln(InfoFile, DUMMY); { FILEBASE }
    {$I-}
    readln(InfoFile, DUMMY);
    if (IOResult = 0) then
      if (DUMMY = 'TRUE') then LUXTRON := True Else LUXTRON := False
    else
      LUXTRON := False;
    {$I+}

  end; { With Status }
  readln(InfoFile, DUMMY);
end; { WriteInfo }


Procedure GetDARSSData(Var DataArray : ScanArray);
var III,jj,kk,Code : integer;
begin
  III := 1;
  While (III <= 1022) do begin
    Read(OptFile, jj);
    for kK := 0 to 7 do
      Read(OptFile,DataArray[III+kK]);
    III := III + 8;
    Readln(OptFile)
  end; { While }
end;


procedure FindBase(Var Analyze:Boolean; var p1, p2 : real);

var p1old, p2old : real;      { Cursor Locations }
   I : Integer;
   dp : Real;
   Good : Boolean;
   s, S1, S2 : String;
```

```
   Point : (Left, Right);
   Ch : Char;
   FS : Integer;

procedure UpDateBaseLine;
begin
  View;
  SetColor(LightBlue);
  SetWriteMode(XORPut);
  MoveTo( ViewX(DARSSParam.Wave[5]   ), ViewY(P1Old) );
  LineTo( ViewX(DARSSParam.Wave[1020]), ViewY(P2Old) );
  MoveTo( ViewX(DARSSParam.Wave[5]   ), ViewY(P1) );
  LineTo( ViewX(DARSSParam.Wave[1020]), ViewY(P2) );
  P1Old := P1;
  P2Old := P2;
end;


begin
  Fs := DARSSParam.DataMax - DARSSParam.DataMin;
  Dp := 1.0;
  Point := Left;
  P1 := 0;
  for i := 5 to 14 do
    P1 := P1 + DARSSParam.PData[i];
  P1 := P1 / 10;
  P1Old := P1;
  P2 := 0;
  For i := 1011 to 1020 do
    P2 := P2 + DARSSParam.PData[i];
  P2 := P2 / 10;
  P2Old := P2;
  if (AutoBase = False) then begin
    SetColor(LightBlue);
    View;
    { Draw BaseLine }
    SetWriteMode(XORPut);
    MoveTo( ViewX(DARSSParam.Wave[5]   ), ViewY(P1) );
    LineTo( ViewX(DARSSParam.Wave[1020]), ViewY(P2) );
    SetWriteMode(CopyPut);
    { Now, modify Baseline As Necessary }
    ViewM;
    ClearViewPort;
    SetColor(White);
    OutTextXY(60,10,'Select BaseLine and Press Return');
    OutTextXY(10,150,'<- Left Point  -> Right Point   PgUp/PgDn : Scale Factor    <ESC> to Skip');
    OutTextXY(10,162,'<Up>, <Down> to Move EndPoint    <Enter> to Accept Base Line');
    Good := false;
    Repeat
      ViewM;
      SetColor(Blue);
      SetWriteMode(CopyPut);
      S := 'End Point: ';
      If (Point = Left) then
```

```
      S := S + 'L'
    else
      S := S + 'R';
    S := S + '    Scale Factor : ';
    Str(Dp:6:2, S1);
    S := S + S1;
    OutTextXY(30,30,S);
    Repeat Until KeyPressed;
    ch := ReadKey;
    if ch = #0 then begin
      ch := ReadKey;
      case ch of
         #75: Point := Left;        { <- }
         #77: Point := Right;       { -> }
         #73: begin                 { PgUp }
            dP := dP + 0.1;
            if (dp > 2) then dp := 2;
                 end;
         #81: begin                 { PgDn }
            dP := dP - 0.1;
            if (dp < 0.1) then dP := 0.1;
            end;
         #72: begin                 { Up }
            if (Point = Left) then
              P1 := P1 + (FS/100)*dP
            else
              P2 := P2 + (FS/100)*dp;
                 UpDateBaseLine;
            end;
         #80: begin                 { Down }
            if (Point = Left) then
              P1 := P1 - (FS/100)*dP
            else
              P2 := P2 - (FS/100)*dp;
            UpDateBaseLine;
                 end;
      end;
    end; {if ch = #0}
    if (ch = #13) then begin
      Good := True;
      Analyze := True;
    end;
    if (Ch = #27) then begin
      Good := True;
      Analyze := False;
    end;
    SetColor(Black);
    OutTextXY(30,30,S);
  Until Good;
 end; { if AUTOBase }
end;


Function SGN(N : Real):Integer;
```

```
begin
  if (N > 0) then
    SGN := 1
  else if (N < 0) then
    SGN := -1
  else
    SGN := 0;
end;


Function Base(i:integer; p1, p2 : real) : Real;
  { Calc. Baseline Value at position I, for given p1 & p2
    (End base line y values) }
begin
  With DARSSParam Do
    base := P1 + (Wave[i]-Wave[5])*(p2-p1)/(Wave[1020]-Wave[5]);
end;


Procedure PeakInfo(var N: integer; p1, p2 : real);
  {
    Based Upon PeakInfo Routine Written for HPLC by
    Timothy J. Dalton
    Stevens Institute of Technology
    CHE416 Final Design Project
    May, 1988
  }
VAR
  Slope0          : Real;        { Initial Slope }
  SlopeN          : Real;
  Slope0Sign      : Integer;     { SGN of Slope0 }
  SlopeNSign      : Integer;
  Slope0Mag       : Real;        { ABS of Slope0 }
  SlopeNMag       : Real;
  Peak            : Real;
  PeakLoc         : Integer;
  PeakStart,
  PeakStop        : Integer;     { Starting Point }
  Area            : Real;        { Peak Area }
  TT              : Real;
  I,J             : Integer;     { Loop Counter }
  Sum,B2,dY       : Real;
  Switch,
  NewPeak,
  SlopeSame,
  SlopeSame2,
  SlopeSame3,
  SlopeSame4      : Boolean;
  Done            : Boolean;
  S,S1            : String;
  OK              : Boolean;


  Procedure OverlappingPeaks;
```

```
  Begin
    OutTextXY(10,100,'OVERLAPPING PEAKS');
  end;

begin
  With DARSSParam do begin
    NumberPeaks := NumberPeaks + 1;
    Switch        := False;
    NewPeak       := True;
    Slope0    := (SData[N] - SData[N-1]) / (Wave[N] - Wave[N-1]);
    Slope0Sign  := SGN(Slope0);
    Slope0Mag   := ABS(Slope0);
    TT     := Wave[N];
    I             := N;
    Repeat
      I := I - 1;
      Sum := 0.0;
      For J := 1 to 5 do
        Sum := Sum + SData[I+J];
      B2 := Sum / 5.0;
      Sum := 0;
      For J := 1 to 5 do
        Sum := Sum + SQR(SData[I+J] - B2);
      dY := Sum / SQRT(5);
    until (dY <= SigmaPeak*FS);
    PeakStart := I;
    Area := 0.0;
    Repeat
      N := N + 1;
      I := N - 5;
      if (I <= 0) then begin
        I := 1;
        OK := True;
        end
      else
        Ok := False;
      Sum := 0.0;
      For J := 1 to 5 do
        Sum := Sum + SData[I+J];
      B2 := Sum / 5.0;
      Sum := 0;
      For J := 1 to 5 do
        Sum := Sum + SQR(SData[I+J] - B2);
      dY := Sum / SQRT(5);
      SlopeN     := (SData[N] - SData[N-1]) / (Wave[N] - Wave[N-1]);
      SlopeNSign  := SGN(SlopeN);
      SlopeNMag   := ABS(SlopeN);
      { Don't Ask me About the Logic on the Next One...I wrote in
       A long Time ago in College... }
      if not( (SlopeNSign=0) and
          ((Slope0Sign=0) or (Slope0Sign=1)) ) then begin

       if (SlopeNSign = Slope0Sign) then
         SlopeSame := True
```

```
   Else
     SlopeSame := False;

   if (Slope0 = SlopeN) then
     SlopeSame3 := True
   Else
     SlopeSame3 := False;

   if (SData[N] > PeakStart*FS) then
     SlopeSame2 := False
   Else
     SlopeSame2 := True;

   if (Wave[N] > (0.1 + TT)) then
     SlopeSame4 := False
   else
     SlopeSame4 := True;

   if ( (SlopeSame  = False) and (SlopeSame2 = False) and
      (SlopeSame3 = False) and (SlopeSame4 = False) and
      (Switch = False) ) then begin
     Switch := True;
     TT    := Wave[N]; end
   else
     if ( (SlopeSame  = False) and (SlopeSame2 = False) and
        (SlopeSame3 = False) and (SlopeSame4 = False) and
        (Switch = True) ) then begin
       Switch  := False;
       NewPeak := False;
       TT     := Wave[N];
     end;

   Slope0    := SlopeN;
   Slope0Sign := SlopeNSign;
   Slope0Mag  := SlopeNMag;

  end;

  if ( (ABS( SData[N] - Base(N,P1,P2)) > PeakEnd*FS) and
         (dY > SigmaPeak*FS) ) then
    Done := False
  else
    Done := True;

  if (OK = True) then
    Done := True;
until Done;

PeakStop := N;
If (NewPeak = False) then OverlappingPeaks;

PStart[NumberPeaks] := PeakStart;
Pend[NumberPeaks] := PeakStop;
Area := 0;
```

```
    Peak := Abs( SData[PeakStart] );
    PeakLoc := PeakStart;
    For J := PeakStart to PeakStop-1 do begin
      Area := Area + ( Wave[J+1]-Wave[J] )
          * ( ( SData[J] - Base(J,P1,P2)) + 0.5 * (SData[J+1]-SData[J]));
      if (Abs( SData[J] ) > Peak) then begin
        Peak := Abs( SData[J] );
        PeakLoc := J;
      end;
    end; { For J }

    PArea[NumberPeaks] := Area;
    PMax[NumberPeaks] := PeakLoc;

    STR( DarssParam.Wave[PeakLoc]:10:4, S);
    S := S + '  ';
    Str(Area*10*DARSSParam.DataMax:10:2, S1);
    S := S + S1;

    SetColor(Yellow);
    OutTextXY(XXPos, YYPos, S);

    YYpos := YYPos + 10;
    if (YYPos > 150)  then begin
      YYPos := 30;
      if (XXPos = 10) then
        XXPos := 150
      else
        if (XXPos = 150) then
          XXPos := 290
        else
          XXPos := 430;
    end;

  end; { With DARSSParam do }
end; { Procedure PeakInfo }

Procedure FindPeaks(p1, p2 : real);
var slope : Real;
begin
  XXPos := 10;
  YYPos := 30;
  SetTextStyle(SmallFont, HorizDir, 4);
  FS := DARSSParam.DataMax-DARSSParam.DataMin;       { Full Scale }
  Fs := 0.1;
  For I := 5 to 1020 do
    DARSSParam.SData[I] := DARSSParam.PData[I]/DARSSParam.DataMax/10;
  P1 := P1/DARSSParam.DataMax/10;
  P2 := P2/DARSSParam.DataMax/10;
  NumberPeaks := 0;
  N := 15;
  While (N <= 1019) Do begin
    if (abs(DARSSParam.SData[n] - Base(n,p1,p2)) > PeakStart*FS) then begin
      slope := (DARSSParam.SData[n] - DARSSParam.SData[n-2])/
```

```
            (DARSSParam.Wave[n] - DARSSParam.Wave[n-2]);

        if ( (abs(slope) > SlopeCriteria*FS ) and
                (NumberPeaks < 50) ) then
          PeakInfo(N,P1,P2);
      end;
        N := N + 1;
    end; { While N <= 1019 Do }
    if (NumberPeaks <> 0) then begin
      Writeln(Out2);
      Writeln(Out2, 'Spectra Number = ',Status.RunNumber:2,'  Time = ',
      TimeG:10:4);
      writeln(out2,'Peak   Start    End     Peak Loc   Peak Area');
      for I := 1 to NumberPeaks do
        with DARSSParam do
          writeln(out2,i:3,' ',Wave[PStart[i]]:10:4,' ',
          Wave[PEnd[i]]:10:4, ' ', Wave[PMax[i]]:10:4, ' ',
          PArea[i]*10*DARSSParam.DataMax:10:4);
    end;
    SetTextStyle(DefaultFont,HorizDir,1);
end;  {Procedure FindPeaks}


Procedure CalcActin;
VAR
  I : Integer;
  J : Integer;
Max  : Real;
Found: Array[1..MaxPeak] of Boolean;

begin
  PArea[NumberPeaks+1] := 0.0;
  For I := 1 to NumPeak do begin
    Found[i] := False;
    J := 1;
    Act[i].PeakCtr := -999;
    While (J <= NumberPeaks) do begin
      if ( act[i].Peak >= (DARSSParam.Wave[Pmax[j]] - Act[i].Delta) ) and
        ( act[i].Peak <= (DARSSParam.Wave[Pmax[j]] + Act[i].Delta) ) then begin
        Act[i].PeakCtr := J;
        J := NumberPeaks + 1;
        Found[i] := True;
      end;
      J := J + 1;
    end;
    if (Found[i] = False) then begin
      Act[i].PeakCtr := NumberPeaks + 1;
      Act[i].WavePeak := 0;
      Max := -1000.0;
      For J := 5 to 1020 do begin
        if (DARSSParam.Wave[j] >= (Act[i].Peak - Act[i].Delta)) and
          (DARSSParam.Wave[j] <= (Act[i].Peak + Act[i].Delta)) then begin
            if (DARSSParam.SData[j] > Max) then begin
                    Max := DARSSParam.SData[j];
```

```
                    Act[i].WavePeak := j;
               end;
             end;
         end;
       end; { Found = False }
     end;
     Write(Out3,TimeG:8:2,' ');
     if (SumOpen = True) then Write(Sum,TimeG:8:2,' ');
     if (NumberPeaks < 50) then begin
       For i := 1 to NumPeak-1 do begin
         if (Parea[Act[NumPeak].PeakCtr] <> 0.0) then begin
               { If Peaks Were Found, we're ok }
           { If no peak, use height ratio }
           if (Act[i].PeakCtr <> (NumberPeaks+1)) then
             Act[i].Ratio := Parea[Act[i].PeakCtr]/PArea[Act[NumPeak].PeakCtr]
           else begin
             Act[i].Ratio := DARSSParam.SData[Act[i].WavePeak] /
                       DARSSParam.SData[Pmax[Act[NumPeak].PeakCtr]];

              writeln(out2,'HEIGHT ACT');
              writeln(out2, DARSSParam.SData[Act[i].WavePeak]);
              writeln(out2, DARSSParam.SData[Pmax[Act[NumPeak].PeakCtr]]);
             end;
           end
         else begin
           Act[i].Ratio := 20000;
           SetColor(RED+BLINK);
           OutTextXY(400,50,'NO ACTIN.!!!');
         end;
         Write(Out3,Act[i].Ratio:9:4,' ');
         if (Sumopen = True) then Write(Sum,Act[i].Ratio:9:4,' ');
         if (act[i].ratio > ACTYMax) then ACTYMax := act[i].ratio;
         if (act[i].ratio < ACTYMin) then ACTYMin := act[i].ratio;
       end; { For i }
       For I:= 1 to NumPeak do begin
         Write(Out3,Parea[Act[i].PeakCtr]*10*DarssParam.DATAMax:12:4);
         if (SumOpen = True) then
               Write(Sum,Parea[Act[i].PeakCtr]*10*DarssParam.DATAMax:12:4);
         if (Parea[Act[i].PeakCtr]*10*DarssParam.DATAMax > RAWYMax) then
           RawYMax := Parea[Act[i].PeakCtr]*10*DarssParam.DATAMax;
         if (Parea[Act[i].PeakCtr]*10*DarssParam.DATAMax < RAWYMin) then
           RawYMin := Parea[Act[i].PeakCtr]*10*DarssParam.DATAMAX
       end;
       Writeln(Out3);
       if (SUmOpen = True) then writeln(Sum);
     end;
   end;


   procedure WriteSData;

   var
     DARSSData : Array[1..1024,1..MaxPerSet] of word;
     J, Code, I, K,
```

```
  NumberFiles              : Integer;
  Dummy, Dum : String;
  Done,SpectraWritten,Ok : Boolean;
  P1, P2 : Real;
  SS: String;
  ReDo: Boolean;

begin
  Status.RunNumber := 0;
  Fname := Status.FileBase;
  Status.FileBase[2] := Status.FileBase[4];
  Status.FileBase[3] := Status.FileBase[5];
  Status.FileBase[4] := Status.FileBase[6];
  Status.FileBase[5] := Status.FileBase[7];
  Status.FileBase[6] := Status.FileBase[8];
  NumberSpectra := 0;
  NumberFiles := 0;
  Done := False;
  TextColor(Red);
  Status.FileBase[0] := Chr(6);
{
  Writeln;
  Writeln('Creating Files: '+Status.Path+'XPS\'+Status.FileBase+'??.XD');
  Writeln;
  Writeln;
  TextColor(Black);
  Writeln('Press Return to BEGIN');
  Repeat Until Keypressed; Resp := Readkey;
}
  TextColor(Black);
  ClrScr;
  GrInit;
  FileName := Status.Path+'OPT\'+FNAME+'.SUM';
  Assign(Out2, FileName);
  ReWrite(Out2);
  Writeln(Out2, 'File = '+FileName);
  Writeln(Out2);
  if (actin) then begin
    Writeln(Out2,'Number of Peaks to Determine is ',NumPeak:2);
    Writeln(Out2);
    For I := 1 to NumPeak do begin
      Writeln(Out2, 'Peak ID is ',Act[i].GasName);
      Writeln(Out2, 'Peak Value is ',Act[i].Peak:10:2,'  Variance is ',
      Act[i].Delta:5:1);
      Writeln(Out2);
    end;
    Writeln(Out2);
  end;
  Assign(Out3,Status.Path+OptPath+FNAME+'.DAT');
  Rewrite(Out3);
  write(out3,' Time   ');
  if (SumOpen = True) then begin
    if (SUmOpen = True) then writeln(sum);
    if (SUmOpen = True) then writeln(sum,'ACTINOMETRY DATA: Ratios and Areas');
```

```
    if (SUmOpen = True) then writeln(sum);
    if (SUmOpen = True) then writeln(sum);
    write(sum,' Time   ');
  end;
  if (actin) then begin
    for I := 1 to NumPeak-1 do begin
      write(out3,'   ',act[i].gasname:4,'   ');
      if (SumOpen = True) then write(Sum,'   ',act[i].gasname:4,'   ');
    end;
    write(out3,'  ');
    if (SumOpen = True) then write(sum,'  ');
    for I := 1 to NumPeak do begin
      write(out3,act[i].gasname:4,act[i].peak:7:1,' ');
      if (SumOpen = True) then
        write(Sum,act[i].gasname:4,act[i].peak:7:1,' ');
    end;
    writeln(Out3);
    if (SumOpen = True) then writeln(Sum);
  end; {If Actin}
  Writeln(Out2);
  While not(eof(OptFile)) and not(done) do begin
    Readln(OptFile, Dummy);
    Readln(OptFIle, Dummy);
    Dum := Copy(Dummy, 1, 5);
    if (Dum <> 'FINAL') then begin
      Readln(OptFile, Dummy);   { Avg Time Note }
      Dummy := Copy(Dummy, 27, 99);
      NumberSpectra := NumberSpectra + 1;
      FileNameGen(Ok);
{
        FileName := Status.Path+'XPS\'+Status.FileBase+'.XD';
        Assign(OutFile,FileName);
}
      GetDARSSData(DARSSParam.Data);
      Repeat
{
        ReWrite(OutFile);
}
        Val(Dummy, TimeG, Code);
        DARSSParam.DataMax := -32768;
        DARSSParam.DataMin := 32767;
        for I := 5 to 1020 do begin
          Temp := (DARSSParam.Data[i]-DARSSParam.BData[i]);
          if (Temp > 32767) then
            Out := Temp - 65536
          else
            Out := Temp;
          DARSSParam.PData[i] := Out;
          if (Out > DARSSParam.DataMax) then
            DARSSParam.DataMax := Out;
          if (Out < DARSSParam.DataMin) then
            DARSSParam.DataMin := Out;
{
          Writeln(OutFile,DARSSParam.Wave[i]:10:4,' ',(Out):5,' ');
```

```
}
            end; { For }
{
        Close(OutFile);
}
        With DARSSParam Do begin
          if (DataMin < -200) then DataMin := -200;
          SetUp(Wave[5], Wave[1022], Wave[1022]-Wave[5],
            DataMin, DataMax, DataMax-DataMin);
          For I := 5 to 1020 do
            if (PData[i] >= DataMin) and (PData[i] <= dataMax) then
              PutPixel( ViewX(Wave[i]), ViewY(PData[i]), Yellow);
        end; { With }

        FindBase(AnalyzeSpectra,p1,p2);

        if ( trunc((P1+P2)/2) < 0.0) then begin
          for I := 5 to 1020 do
            DARSSParam.Data[i] := DARSSParam.Data[i] - trunc((P1+P2)/2);
          ReDo := True; end
        else
          ReDo := False;

      Until Not(ReDo);

    {COMMENT HERE}
      If ( (AnalyzeSpectra = True) or (AutoBase = True) ) then begin
        ViewM;
        ClearViewPort;
        SetColor(Red);
        OutTextXY(10,10,'FIND PEAKS');
        Str(TimeG:10:4,SS);
        OutTextXY(150,10,'Time   : '+SS);
        Str(NumberSpectra:2,SS);
        OutTextXY(150,20,'Number : '+SS);
        { Plot the Fucking Spectra }
        if (PlotSpectra = 'Y') then begin
          SSEQ := SSEQ + 1;
writeln('CALLING PLOT THIS SPECTRUM');
                  PlotThisSpectra(SSEQ);
        end;
        { Is it a Valid Spectra on Auto Mode ??? }
        if ( (DARSSParam.DataMax-DARSSParam.DataMin > 400)
                and (AutoBase = True) ) then begin
          If (Actin = True) then begin
            FindPeaks(P1,P2);
                    CalcActin;
          end;
        end;
      end;
      ViewM;
      SetColor(White);
      if (AutoBase = False) then begin
        CenterTextX(MXE-MXS, 100, 'PRESS Return');
```

```
        Repeat Until KeyPressed;
        Resp := ReadKey;
        ClearViewPort;
      end;
    {COMMENT HERE}

    end; {IF DUM <> FINAL}
  end; { While }
  If (Actin = True) then begin
    ASEQ := ASEQ + 1;
    PlotActin(ASEQ);
    PlotRaw(ASEQ);
  end;
  Close(OptFile);
  Close(Out2);
  Close(Out3);
  Status.FileBase := Fname;
  CloseGraph;
  TextBackGround(LightGray);
  ClrScr;
end; { Procedure }


Procedure CenterText(Y:integer; stuff:string);
Var
  X : Integer;

begin
  X := (80 - Length(Stuff)) div 2;
  GotoXY(X,Y); Write(Stuff);
end;


Procedure GetActinParam;
var I : integer;
begin
  ClrScr;
  TextColor(Blue);
  Writeln('Enter Number of Peaks to Find (- for a Standard Set)');
  Writeln('Set -1:  HBr/Clr 7250 A Center' );
  Writeln;
  Write('Enter Number : ');
  TextColor(Black);
  if (not(Auto)) then
    Readln(NumPeak)
  else
    NumPeak := -1;
  writeln;
  if (NumPeak > 0) then begin
    Writeln('Enter Peak Info, Actinometer is the FINAL PEAK');
    For I := 1 to NumPeak do begin
      if (I = NumPeak) then
        Writeln('ENTER THE ACTINOMETER INFORMATON');
      Write('Enter Peak #',i:2,' Name '); Readln(Act[i].GasName);
```

```
      Write('Enter Peak #',i:2,' Peak Location (A) '); Readln(Act[i].Peak);
      Write('Enter Peak #',i:2,' Peak Variance (A) '); Readln(Act[i].delta);
      writeln;
    end;
    ClrScr; end
  else
    begin
      NumPeak := 5;
      Act[1].GasName := 'Br';
      Act[1].Peak := 7005;
      Act[1].Delta := 5;
      Act[2].GasName := 'Br';
      Act[2].Peak := 7349;
      Act[2].Delta := 10;
      Act[3].GasName := 'Cl';
      Act[3].Peak := 7414;
      Act[3].Delta := 5;
      Act[4].GasName := 'Cl';
      Act[4].Peak := 7547;
      Act[4].Delta := 5;
      Act[5].GasName := 'Ar';
      Act[5].Peak := 7505;
      Act[5].Delta := 5;
    end;
end;

Procedure PeakFix;
Var S : Str8; Code : Integer;
begin
  ClrScr;
  S := '0.035';
  TextColor(Blue);
  Write('Enter  PeakStart  Value (0.01 to 0.1 recommended)  : ');
  TextColor(Black);
  GetString(S, Numeric, WhereX, WhereY);
  Val(S, PeakStart, Code); Writeln;
  S := '0.040';
  TextColor(Blue);
  Write('Enter  PeakEnd  Value (0.01 to 0.03 recommended)   : ');
  TextColor(Black);
  GetString(S, Numeric, WhereX, WhereY);
  Val(S, PeakEnd, Code); Writeln;
  S := '0.00005';
  TextColor(Blue);
  Write('Enter  SigmaPeak  Value (5e-4 to 4e-6 recommended) : ');
  TextColor(Black);
  GetString(S, Numeric, WhereX, WhereY);
  Val(S, SigmaPeak, Code); Writeln;
  S := '0.01';
  TextColor(Blue);
  Write('Enter  SlopeCriteria  Value (0.001 to 0.1 recommended) : ');
  TextColor(Black);
  GetString(S, Numeric, WhereX, WhereY);
  Val(S, SlopeCriteria, Code); Writeln;
```

```
end; {Procedure PeakFix}


Procedure LiferFix;
Var S : Str8; Code : Integer;
begin
  S := '1.0';
  TextColor(Blue);
  Write('Enter Index for Medium Ouside the Film, N1 : ');
  TextColor(Black);
  GetString(S, Numeric, WhereX, WhereY);
  Val(S, N1, Code); Writeln;
  S := '3.7';
  TextColor(Blue);
  Write('Enter Index for Underlying Film, N2 : ');
  TextColor(Black);
  GetString(S, Numeric, WhereX, WhereY);
  Val(S, N2, Code); Writeln;
  S := '632.8';
  TextColor(Blue);
  Write('Enter Laser WaveLength in nM : ');
  TextColor(Black);
  GetString(S, Numeric, WhereX, WhereY);
  Val(S, Lambda, Code); Writeln;
  S := '5.0';
  TextColor(Blue);
  Write('Enter Laser Beam Angle WRT Surface Normal in Deg. : ');
  TextColor(Black);
  GetString(S, Numeric, WhereX, WhereY);
  Val(S, Theta, Code); Writeln;
  S := 'Y';
  TextColor(Blue);
  Write('Calculate Etching Rate ? (Y/N) : ');
  TextColor(Black);
  GetString(S, Alfa, WhereX, WhereY);
  S := UpCase(S[1]);
  If (S = 'Y') then
    CalcER := TRUE
  else
    CalcER := False;

end; {Procedure}


Procedure SEQCOUNT;
{
  Count the AME Plot Sequences!

  This Subroutine examines the default path subdirectory PROPLOT
  for .pl files until its ok.
}


Var
```

```
  FileName,filenm : String;
  DirInfo  : SearchRec;     { PreDefined in DOS Unit }

begin
  LSEQ := 0;
  Repeat
    LSEQ := LSEQ + 1;
    GetFileName(FileNm, 'L', LSEQ);
    FileName := status.path+PlotPath+filenm+'.pl';
    FindFirst(FileName, AnyFile, DirInfo);
  Until (DosError <> 0);
  Lseq := Lseq -1;

  SSEQ := 0;
  Repeat
    SSEQ := SSEQ + 1;
    GetFileName(FileName, 'S', SSEQ);
    FileName := status.path+PlotPath+filename+'.pl';
    FindFirst(FileName, AnyFile, DirInfo);
  Until (DosError <> 0);
  Sseq := Sseq -1;

  ASEQ := 0;
  Repeat
    ASEQ := ASEQ + 1;
    GetFileName(FileName, 'R', ASEQ);
    FileName := status.path+PlotPath+filename+'.pl';
    FindFirst(FileName, AnyFile, DirInfo);
  Until (DosError <> 0);
  Aseq := ASeq - 1;

  ESEQ := 0;
  Repeat
    ESEQ := ESEQ + 1;
    GetFileName(FileName, 'I', ESEQ);
    FileName := status.path+PlotPath+filename+'.pl';
    FindFirst(FileName, AnyFile, DirInfo);
  Until (DosError <> 0);
  Eseq := Eseq - 1;

end; {Procedure}

begin
  LSEQ := 0;
  SSEQ := 0;
  ASEQ := 0;
  ESEQ := 0;
  TSEQ := 0;


  AUTO := False;
  CountRuns(NoRun);

  SEQCOUNT;
```

```
if (NoRun = 0) then begin
  TextBackGround(LightGray);
  TextColor(RED);
  ClrScr;
  Status.FileBase[0] := Chr(6);
  CenterText(5,'NO RUNS FOUND for '+Status.FileBase);
  TextColor(Black);
  CenterText(10,'Press ENTER to Continue');
  Repeat Until KeyPressed;
  Resp := ReadKey;
end;

TextBackGround(LightGray);
TextColor(RED);
ClrScr;
Writeln;
CenterText(10,'Press ENTER to BEGIN');
CenterText(12,'A for Auto, R for Starting Run Number');
Repeat Until KeyPressed;
Resp := ReadKey;
if (Resp = 'A') then begin
  Auto := True;
end;
if ((Resp = 'R') or (Resp = 'A') ) then begin
  writeln;
  writeln('Enter Starting Run Number:');
  readln(StartRun)
  end
else begin
  Auto := False;
  StartRun := 1;
end;

for II := StartRun to NoRun Do Begin
  TextBackGround(LightGray);
  TextColor(RED);
  ClrScr;
  Status.FileBase[0] := chr(6);
  Str(NoRun:2,RunString);
  CenterText(1,'File Series '+Status.FileBase+' contains '+RunString+' runs.');
  Writeln;
  TextColor(Blue);
  MakeFileName(II);
  CenterText(3,'File: '+Status.FileBase);
  FileName := Status.Path+InfoPath+Status.FileBase+'.INF';
  Assign(InfoFile, FileName);
  ReSet(InfoFile);
  ReadInfo;
  ShowInfo;
  TextColor(Blue);
  writeln; writeln; writeln;
  write('Plot ? : ');
  if (Not(Auto)) then
    Readln(plotIt)
```

```
else
  PlotIt := 'Y';
plotit := UpCase(plotit);
Writeln;
SumOpen := False;

if (plotit = 'Y') then begin

  if (SumOpen= True) then begin
    Close(Sum);
    SumOpen := False;
  end;
  assign(sum, Status.Path+InfoPath+Status.FileBase+'.SM');
  rewrite(sum);
  SumOpen := True;

  if (status.lifer = true) then begin
    TextColor(Blue);
    Write('Modify Interferometry Parameters ? (Y)es (P)oly (O)xide (N)one: ');
    TextColor(Black);
    if(Not(Auto)) then
         Readln(Resp)
    else
      Resp := 'P';

        Resp := UpCase(Resp);

   case resp of
   'Y' : LiferFix;
   'P' : begin
        { Set Up Defaults for Poly}
        n1     := 1.0;
            n2     := 3.7;
            lambda := 632.8;
            theta  := 5.0;
        CalcER := True;
        end;
  'O'  : begin
        {Set Up Defaults for Oxide}
        n1     := 1.0;
        n2     := 1.45;
        lambda := 632.8;
        theta  := 5.0;
        CalcER := True;
        end;
 'N'  : CalcER := False;
 else
   CalcER := False;

 end;

 gettimes;

 lseq := lseq + 1;
```

```
        PlotLifer(LSEQ);
  end; {Status.LIFER}

  WriteSum;

  if (status.thr640) then begin
    Sseq := Sseq + 1;
    if (Status.THR640PMT.WaveScan) then
      PlotTHR640Wave(Sseq)
    else begin
          TextColor(Blue);
      Write('Filter Data (Low Pass) ? (Y/N) ');
      TextColor(Black);
      if(Not(Auto)) then
            Readln(Resp)
      else
        Resp := 'N';
      Resp := UpCase(Resp);
      if (Resp = 'Y') then begin
            DataFilter := True;
            TextColor(Blue);
        Write('Enter Number of Times to Filter: ');
        TextColor(Black);
        if(Not(Auto)) then
              Readln(NumFilter)
        else
          NumFilter := 1;
        end
      else
         DataFilter := False;
      PlotTHR640OptEmm(Sseq);
    end;
  end; {Status.Thr640}

  if (status.comdel = true) then begin
    { Determine if its OLD data format or NEW
     data format }
    Assign(f,status.path+IVPath+status.filebase+'.iv');
        reset(f);
        Readln(f,tempstr);
    close(f);
    eseq := eseq + 1;
    if ((tempstr[1] = 'V') and (tempstr[2] = '2')) then begin
      PlotElec1(eseq);
      PlotElec2(eseq) end
    else begin
      PlotElec1OLD(eseq);
      PlotElec2OLD(eseq);
    end;

  end;  { Status.Comdel }

end; { If Plotit!}
```

```
if (Status.DARSS) then begin
  TextColor(Blue);
      Writeln; Writeln('Grating : ',DARSSParam.Grating:1,
  '         Grating Center : ',DARSSParam.Center:8:1,' A');
  Writeln('Exposure: ',DARSSParam.Exposure:5,' ms      ',
  'Correction    : ',DARSSParam.Correction:8:1,' A');
  Writeln;
  Write('Reprocess ? : '); textcolor(Black);
  if (not(Auto)) then
    Readln(Resp)
  else
    Resp := 'Y';

      Resp := UpCase(Resp);
  Writeln;
  if (Resp = 'Y') then begin
    TextColor(Blue);
    Writeln; Write('Plot Spectra ? '); textcolor(Black);
    if (not(Auto)) then
      Readln(PlotSpectra)
    else
      PlotSpectra := 'Y';

        PlotSpectra := UpCase(PlotSpectra); TextColor(Blue);
    Writeln; Write('Automate Baseline Selection ? '); textcolor(Black);
    if (Not(Auto)) then
      Readln(Resp)
        else
          Resp := 'Y';

        Resp := UpCase(Resp);
    If (Resp = 'Y') then
      AutoBase := True
    Else
      AutoBase := False;

    TextColor(Blue);
    Writeln; Write('Calculate Actinometry? '); textcolor(Black);
    if (not(Auto)) then
      Readln(Resp)
        else
          Resp := 'Y';

        Resp := UpCase(Resp);
    If (Resp = 'Y') then
      Actin := True
    Else
      Actin := False;

    if (Actin) then GetActinParam;

    TextColor(Blue);
    Writeln; Write('Modify Peak Location Parameters ? ');
    TextColor(Black);
```

```
      if (not(Auto)) then
        Readln(Resp)
          else
            Resp := 'N';

          Resp := UpCase(Resp);
      If (Resp = 'Y') then
        PeakFix
      else begin
        PeakStart := 0.035;
        PeakEnd := 0.040;
        SigmaPeak := 0.00005;
        SlopeCriteria := 0.01;
      end;

      InitCalib;
      CalcWave;
      FileName := Status.Path+OptPath+Status.FileBase+'.Opt';
          Assign(OptFile, FileName);
      ReSet(OptFile);
      Readln(OptFIle, Dummy); Readln(OptFIle, Dummy);
      GetDARSSData(DARSSParam.BData);
      ACTYMax := -99999;
      ACTYMin := 99999;
      RAWYMax := -99999;
      RAWYMin := 99999;
      WriteSData
    end; {If Resp = 'Y'}
  end; {Status.DARSS}

  if (Status.Luxtron) then begin
   TSEQ := TSEQ + 1;
       PlotLuxtron(TSEQ);
  end;

  Writeln;
  Close(InfoFile);

  if (SumOpen= True) then begin
    Close(Sum);
    SumOpen := False;
  end;
 end; {For II}
end.
```

```
UNIT MsOPTT;

INTERFACE
Uses Dos, Crt, Vars, Misc;
Procedure Interrogate;
Procedure ShowInfo;

Implementation

Procedure Interrogate;
Var
  Ch:String; Cha : Char;
  X,Y : Byte;
  Year, Month, Day, DayOfWeek : Word;
  Good,done,Happy : Boolean;
  I : Integer;
begin
  Happy := false;
  WITH Status Do Begin
    RunNumber := 0;
    Files := Closed;
    TextBackGround(LightGray);
    ClrScr;
    TextColor(Red+Blink);
    CenterText(1,'A.M.E. 5000  Automated Data Analysis System');
    TextColor(Blue);
    GotoXY(1,4); Write('Enter 6 Digit Run Sequence : ');
    TextColor(Black);
    WriteBlanks(10,x,y);
    EnterText(Ch, X, Y, 6);
    for i := 1 to 6 do
      FileBase[i] := UpCase(Ch[i]);
    FileBase[0] := Chr(6);
    GotoXY(X,Y); Write(FileBase);
    TextColor(Blue);
    GotoXY(1,6); Write('Enter File Path Name : ');
    TextColor(Black);
    WriteBlanks(42,x,y);
    EnterText(Ch, X, Y, 40);
    For I := 1 to Ord(Ch[0]) Do
      Path[I] := UpCase(Ch[I]);
    Path[0] := Ch[0];
    If (Path[ Ord(Path[0]) ] <> '\') then
      begin
        Inc( Path[0] );
        Path[ Ord(Path[0]) ] := '\';
      end;
    GotoXY(X,Y);  Write(Path);
  end; {WITH}
end; { Procedure Interrogate }


procedure ShowInfo;
begin
```

```
TextColor(Red);
CenterText(5,'A C Q U I S I T I O N   P A R A M E T E R S');
TextColor(Blue);
GotoXY(5,7); write('Laser InterFerometry   : :');
GotoXY(5,8); write('THR640 WaveLength Scan  : :');
GotoXY(5,9); write('THR640 Optical Emission : :');
GotoXY(5,10); write('Luxtron Temperature   : :');
GotoXY(45,7); Write('THR640 Diode Array     : :');
GotoXY(45,8); Write('DARSS Diode Array      : :');
GotoXY(45,9); Write('Comdel RPM             : :');
TextBackground(Cyan);
with Status Do Begin
  TextColor(red);
  GotoXY(30,7);
  if LIFER Then
        write('X')
  else
      write(' ');
  GotoXY(30,8);
  if THR640PMT.WaveScan Then
        write('X')
  else
      write(' ');
  GotoXY(30,9);
  if THR640PMT.OptEmission Then
        write('X')
  else
      write(' ');
  GotoXY(30,10);
  if LUXTRON then
              write('X')
  else
              write(' ');
  GotoXY(70,7);
  if THR640LDA Then
        write('X')
  else
      write(' ');
  GotoXY(70,8);
  if DARSS Then
        write('X')
  else
      write(' ');
  GotoXY(70,9);
  if COMDEL Then
        write('X')
  else
      write(' ');
 end; {With }
 TextBackGround(LightGray);

(*
   if (THR640PMT.WaveScan) or (THR640PMT.OptEmission) then
     begin
```

```
      PMT := True;
      THR640 := True;
    end;
  if (THR640LDA) then begin
    LDA := True;
    THR640 := True;
  end;
  { Should We Read Input for The Grating ? }
  if (thr640) then
    Good := False
  else
    Good := True;
  TextBackGround(LightGray);
  GotoXY(1,19); for i := 1 to 74 do write(' ');
  While not(good) do begin
    TextColor(Blue);
    GotoXY(1,19); Write('Enter Grating to Use (A/B) : ');
    TextColor(Black);
    WriteBlanks(3,x,y);
    EnterText(Ch, X, Y, 1);
    Grating := UpCase(Ch[1]);
    If (Grating = 'A') or (Grating = 'B') then
      begin
        Good := True;
        GotoXY(X,Y);
        Write(Grating);
      end
    Else
      Good := False;
  end; {While}
  { Should We Read Input for The DARSS Grating ? }
  if (DARSS) then
    Good := False
  else
    Good := True;
  TextBackGround(LightGray);
  GotoXY(1,19); for i := 1 to 74 do write(' ');
  While not(good) do begin
    TextColor(Blue);
    GotoXY(1,19); Write('Enter the Grating (1-3) on the Jarrel-Ash : ');
    TextColor(Black);
    WriteBlanks(3,x,y);
    EnterText(Ch, X, Y, 1);
    DARSSParam.Grating := ord(Ch[1]) - ord('0');
    Ch[1] := UpCase(Ch[1]);
    If (Ch[1] = '1') or (Ch[1] = '2') or (Ch[1] = '3') then
      begin
        Good := True;
        GotoXY(X,Y);
        Write(Ch[1]);
      end
    Else
      Good := False;
    GotoXY(1,20);
```

```
      TextColor(Blue);
      Write('Enter Meter Reading from Jarrel-Ash in Angstroms : ');
      TextColor(Black);
      WriteBlanks(10,x,y);
          Readln(DARSSParam.Center);
      TextColor(Blue);
      Write('Enter Exposure Time in ms : ');
      TextColor(Black);
      WriteBlanks(10,x,y);
          Readln(DARSSParam.Exposure);
      TextColor(Blue);
      Write('Enter Correction Factor in Angstroms : ');
      TextColor(Black);
      WriteBlanks(10,x,y);
          Readln(DARSSParam.Correction);
    end; {While}
  end; {WITH}
    Repeat
      TextColor(Blue);
      GotoXY(1,24); Write('Are All These Correct? (Y/N) : ');
      TextColor(Black);
      WriteBlanks(3,x,y);
      EnterText(Ch, X, Y, 1);
      Cha := UpCase(Ch[1]);
      If (Cha = 'Y') or (Cha = 'N') then
        begin
          Good := True;
          GotoXY(X,Y);
          Write(Cha);
        end
      Else
        Good := False;
    Until Good;
    if (Cha = 'Y') then
      Happy := true
    else
      Happy := False;
  Until Happy;

*)

end; {Procedure Show Info}


begin
end.
```

```
unit MSOptG;
{
        The Constants in This Section are Defined for a ***VGA***
     Screen.  The output will be ***INCORRECT*** on other screen types
}

interface

Uses Graph, Dos, Crt, Misc, VARS;

const
  { Define Graph 1 & 2  X&Y Start and End Points }
  GrXS  : integer = 0;
  GrYS  : integer = 15;
  GrXE  : integer = 639;
  GrYE  : integer = 300;
  MXS   : integer = 0;
  MXE   : integer = 639;
  MYS   : integer = 301;
  MYE   : Integer = 479;

  Grid  : Boolean = True;

var graphdriver, graphmode: integer;
   xm,ym : integer;
   vxs, vys, vxe, vye : integer;
   SizeX, SizeY : integer;
   MinX, MaxX, dx, MinY, MaxY, dy: real;
   XViewMin, XViewMax, YViewMin, YViewMax : integer;

   { List All The Procedures }
   procedure CenterTextX(xmax, y:integer; s:string);
   procedure CenterTextY(x, ymax:integer; s:string);
   function  ViewY(Y:real):integer;
   function  ViewX(X:real):integer;
   procedure GrInit;
   procedure GrView(MinXL, MaxXL, dXL, MinYL, MaxYL, dYL: Real);
   Procedure View;
   Procedure ViewM;
   procedure SetUp(MinX, MaxX, dX, MinY, MaxY, dY: Real);

Implementation

procedure CenterTextX(xmax, y:integer; s:string);
{ Center a String in the X Direction }
begin
 OutTextXY(((xmax - TextWidth(s)) div 2),y,s);
end;

procedure CenterTextY(x, ymax:integer; s:string);
{ Center a String in the Y Direction }
begin
 OutTextXY(x,((ymax - TextWidth(s)) div 2),s);
end;
```

```
{ Convert From Window Coordinates(real) to Viewport Coordinates(Int) }
function ViewY(Y:real):integer;
var m : real;
begin
 m := (YViewMax - YViewMin) / (MaxY - MinY);
 ViewY := YViewMin + round(m * (Y-MinY));
end;

function ViewX(X:real):integer;
var m : real;
begin
 m := (XViewMax - XViewMin) / (MaxX - MinX);
 ViewX := XViewMin + round(m * (X-MinX));
end;

procedure GrInit;
{Initialize Graphics}
begin
 graphdriver := detect;
 InitGraph(graphdriver, graphmode, '');
 if (graphresult <> grok) then
   halt(2);
 ClearDevice;
 xm := GetMaxX;
 ym := GetMaxY;
 SetColor(White);
 Rectangle(0,0,Xm,Ym); { Outline Entire Screen }
 MoveTo(0,15); LineTo(xm,15); { Draw a Line to Separate Titles }
 SetColor(LightRed);
 Rectangle(GrXS, GrYS, GrXE, GrYE); { 1st Graph Boundry }
end;

procedure View;
begin
  SetViewport(vxs, vys, vxe, vye, ClipOn);
end;


procedure ViewM;
begin
  SetViewport(mxs+1, mys+1, mxe-1, mye-1, ClipOn);
end;

procedure GrView(MinXL, MaxXL, dXL, MinYL, MaxYL, dYL: Real);
begin
  { Calc. Allowed Viewport For Graph 1 }
  vxs := GrXS + 1;
  vys := GrYS + 1;
  vxe := GrXE - 1;
  vye := GrYE - 1;
  { Calc. Size of Window in Pixels }
  SizeX := vxe - vxs;
  SizeY := vye - vys;
  { Store Calling Parameters }
```

```
  MinX := MinXL; MaxX := MaxXL; dx := dxL;
  MinY := MinYL; MaxY := MaxYL; dy := dyL;
  SetViewport(vxs, vys, vxe, vye, ClipOn);
  ClearViewPort;
  { Calc. Graph Drawing Limits Leaving Rooms For Axes & Labels }
  XViewMin := 0 + (SizeX div 10);
  YViewMax := 0 + (SizeY div 10);
  XViewMax := SizeX - ((SizeX * 4) div 100);
  YViewMin := SizeY - (SizeY div 8);
  { Rectangle(XView2Min, YView2Min, XView2Max, YView2Max);}
end;

procedure SetUp(MinX, MaxX, dX, MinY, MaxY, dY: Real);
{ Given Real World Coordinate Limits, Draw Graph, including axes,
 labels for numbers and titles }
var a,b,c,d,XStart,YStart : integer; s,s1 : string; e: real;

function XAxisVisible:boolean;
begin
 if (MinY <= 0) and (MaxY >= 0) then
   XAxisVisible := True
 else
   XAxisVisible := False;
end;

function YAxisVisible:boolean;
begin
 if (MinX <= 0) and (MaxX >= 0) then
   YAxisVisible := True
 else
   YAxisVisible := False;
end;

begin
  SetTextStyle(SmallFont, HorizDir, 4);
  GrView(MinX, MaxX, dX, MinY, MaxY, dY);

  if XAxisVisible then
    YStart := ViewY(0.0)
  else
    YStart := YViewMin;
  a := ViewX(MinX);
  b := ViewX(MaxX);
  SetColor(White);
  c := YStart;
  MoveTo(a,c);
  LineTo(b,c);
  e := MinX;
  while e <= MaxX do begin
    b := ViewX(e);
    MoveTo(b,c);
    d := SizeY div 50;
    SetColor(White);
    LineTo(b,(c+d));
```

```
    str(e:5:1,s);
    SetColor(LightCyan);
    OutTextXY( b-TextWidth(s) div 2,c+2*d,s);
    if grid then begin
      SetColor(LightGray);
      MoveTo(b,YViewMin);
      LineTo(b,YViewMax);
    end;
    e := e + dx;
  end;
  SetColor(Brown);
  CenterTextX(sizex,SizeY-15,'WaveLength(A)');

  if YAxisVisible then
    XStart := ViewX(0.0)
  else
    XStart := XViewMin;
  a := ViewY(MinY);
  b := ViewY(MaxY);
  SetColor(White);
  c := XStart;
  MoveTo(c,a);
  LineTo(c,b);
  e := MinY;
  while e <= MaxY do begin
    b := ViewY(e);
    MoveTo(c,b);
    d := SizeX div 100;
    SetColor(White);
    LineTo((c-d),b);
    str(e:5:1,s);
    SetColor(LightCyan);
    OutTextXY( c-2*d-TextWidth(s),b-TextHeight('A') div 2,s);
    if grid then begin
      SetColor(LightGray);
      MoveTo(XViewMin,b);
      LineTo(XViewMax,b);
    end;

    e := e + dy;
  end;
  SetTextStyle(DefaultFont,VertDir,1);
  SetColor(Brown);
  CenterTextY(14,SizeY,'Intensity');
  SetTextStyle(DefaultFont,HorizDir,1);

end;


end.
```

```
unit plot;

{ Generate ProPlot Files }

INTERFACE
Uses Dos, Crt, Vars, Misc, MSOPTVG, DT2811TL;

PROCEDURE GetFileName(VAR Filenm:String; initial:string; Sequence:Integer);
PROCEDURE PlotLifer(Sequence:Integer);
PROCEDURE PlotLuxtron(Sequence:Integer);
PROCEDURE PlotThR640OptEmm(Sequence:Integer);
PROCEDURE PlotThR640Wave(Sequence:Integer);
PROCEDURE PlotThisSpectra(Sequence:Integer);
PROCEDURE PlotActin(Sequence:Integer);
PROCEDURE PlotRAW(Sequence:Integer);

PROCEDURE WriteSum;

IMPLEMENTATION

PROCEDURE GetFileName(VAR Filenm:String; initial:string; Sequence:Integer);

VAR S:String;
begin
  Filenm := 'AME'+initial;
  STR(Sequence:4, S);
  if s[1] = ' ' then s[1] := '0';
  if s[2] = ' ' then s[2] := '0';
  if s[3] = ' ' then s[3] := '0';
  Filenm := Filenm + s;
end; {Procedure GetFileName}


PROCEDURE PlotLifer(Sequence:Integer);
{ Produce the Initial Plot }
var xlim,
   t, v    : real;
   f,f2    : text;
   filenm,
   dummy          : string;
   a              : char;
   delta  : real;
   prevmax,
   prevmin        : real;
   EtchTime       : real;
   EtchingRate : Real;
   EtchingRatePiece : Real;
   EtchingBase : Real;
   NumEtch        : Integer;
   I,J            : Integer;

 begin

   a := chr(39);
```

```
assign(f,Status.path+LiferPath+status.filebase+'.LIN');
reset(f);
NumLifPts := 0;
while not(eof(f)) do begin
  readln(f, t,v);
  NumLifPts := NumLifPts + 1;
end;
RunTime := t;
endtime := 10.0*int(t/10.0 +1.0);          { Round it up to next mult. of 10 }
close(f);
GetFileName(filenm,'L',sequence);
assign(f,status.path+PlotPath+filenm+'.pl');
rewrite(f);
writeln(f,'%     Laser Interferomety Automated Plot Generator');
writeln(f,'      '+status.path+liferpath+status.filebase+'.lin');
writeln(f,'%     Timothy J. Dalton (c) 1991');
writeln(f);
writeln(f,'set page format landscape');
writeln(f,'set window x 1.5 to 9.5 y 3 to 7');
writeln(f,'set limits x 0 ',endtime:4);
writeln(f,'title bottom '+a+'Time (Seconds)'+a);
writeln(f,'title top '+a+status.filebase+a);
writeln(f,'title left '+a+'Laser Interferometry Signal (V)'+a);
if (numlifpts > 1024) then begin
  I := round(numlifpts);
  j := I div 1024;
  writeln(f,'set sample ',(j+1):2);
end;
writeln(f,'insert '+status.path+liferpath+status.filebase+'.lin');
writeln(f,'join');
writeln(f,'set legend spacing 1.2 justify left size 1.0');
Assign(f2, Status.Path+InfoPath+Status.FileBase+'.INF');
ReSet(f2);
readln(f2, Dummy);
writeln(f,'legend 0.5 2.0 '+a+dummy+a);
{ Read in pre run comments }
Repeat
  Readln(f2,Dummy);
  writeln(f, 'legend '+a+dummy+a);
  if (SumOpen = True) then writeln(sum,dummy);
Until (Dummy[1] = '*') and (Dummy[0] > #1);
{ Skip the run Info }
repeat
  readln(f2,dummy);
Until (Dummy[1] = '*') and (Dummy[0] > #1);
{ Skip the darss times if there... }
repeat
  readln(f2,dummy);
Until (Dummy[1] = '*') and (Dummy[0] > #1);
writeln(f,' legend 4.0 2.0 '+a+dummy+a);
repeat
  Readln(f2,Dummy);
  writeln(f, 'legend '+a+dummy+a);
until eof(f2);
```

```
 close(f2);
 writeln(f,'legend 9.6, 7, '+a+'LIFER'+a);
 if (status.darss) then begin
   writeln(f,'legend '+a+'DARSS'+a);
   writeln(f,'legend '+a+'Grating : ',darssParam.grating:1,a);
   writeln(f,'legend '+a+'Center  : ',darssParam.center:5:0,a);
   writeln(f,'legend '+a+'Exposure: ',darssParam.exposure:5,a);
   writeln(f,'legend '+a+'Correc. : ',darssParam.correction:5:1,a);
 end;
 if (status.comdel) then begin
   writeln(f,'legend '+a+'COMDEL'+a);
 end;
 if (status.luxtron) then begin
   writeln(f,'legend '+a+'LUXTRON'+a);
 end;
 if (status.thr640) then begin
   writeln(f,'legend '+a+'THR640'+a);
   if (status.thr640lda) then begin
     writeln(f,'legend '+a+'LDA'+a);
   end;
   if (status.thr640pmt.WaveScan) then begin
     writeln(f,'legend '+a+'PMT Wavescan'+a);
   end;
   if (status.thr640pmt.OptEmission) then begin
     writeln(f,'legend '+a+'PMT OptEmission'+a);
   end;
 end;
end;
if (NumberExtrema <> 0)  then begin
  EtchTime := 0.0;
  NumEtch := 0;
  writeln(f,'set legend spacing 1.2 justify left size 0.85 font courier');
  writeln(f,'legend 7.75 2.15 '+a+'EXTREMA'+a);
  writeln(f,'legend 7.5 2.0 '+a+'    Times Type Delta  Etching R.'+a);
  prevmax := 0.0;
  theta := theta * 3.141592654 / 180.0;
  EtchingBase := (Lambda/2.0) / sqrt(n2*n2 - n1*n1*sin(theta)*sin(Theta));
  EtchingRate := 0.0;
  prevmin := 0.0;
  for I := 1 to NumberExtrema do begin
    if (ExtremaType[i] = 'Max') then begin
      delta := extrematimes[i] - prevmax;
      prevmax := extrematimes[i];
      end
    else begin
      delta := extrematimes[i] - prevmin;
      prevmin := extrematimes[i];
    end;
    EtchingRatePiece := EtchingBase/delta*60;
    if ((I >= 3) and (I <= NumberExtrema-2)) then begin
      EtchTime := EtchTime + delta;
      EtchingRatePiece := EtchingBase/delta*60;
      EtchingRate  := EtchingRate + EtchingRatePiece;
      NumEtch := NumEtch + 1;
    end;
```

```
      if (i <> 15) then
        writeln(f,'legend '+a,ExtremaTimes[i]:10:3,' ',
                 ExtremaType[i],' ',delta:5:2,' ',
                     EtchingRatePiece:10:2,a)
      else
        writeln(f,'legend 9.0 2.0 '+a,ExtremaTimes[i]:10:3,' ',
                 ExtremaType[i],' ',delta:5:2,' ',
                     EtchingRatePiece:10:2,a);
    end; {For I}

    for I := 1 to NumberExtrema do
      writeln(f,ExtremaTimes[i]:10:3,' ',0.0);
    writeln(f,'set symbol 2D');
    writeln(f,'plot');
    etchingRate := EtchingRate / NumEtch;
    writeln(f,'legend 9.6 4 '+a+'Etching Rate (3 --> ',
         (NumberExtrema-2):2,')'+a);
    writeln(f,'legend '+a+'l  = ',lambda:5:1,' nM',a);
    writeln(f,'case '  +a+'G            '+a);
    writeln(f,'legend '+a+'N2 = ',n2:5:2,a);
    writeln(f,'legend '+a+'N1 = ',n1:5:2,a);
    writeln(f,'legend '+a+'Q  = ',theta:6:3,' Deg.',a);
    writeln(f,'case '  +a+'G          '+a);
    writeln(f,'legend '+a+'ER = ',(EtchingRate):10:2,' nM/Min',a);

  end; {if NumberExtrema}
  writeln(f,'stamp bottom date time');
  close(f);
end;


PROCEDURE PlotLuxtron(Sequence:Integer);
{ Produce the Initial Plot }
var xlim,
   t, v1,v2          : real;
   f,f2    : text;
   filenm,
   dummy             : string;
   a                 : char;
   delta   : real;
   prevmax,
   prevmin           : real;
   EtchTime          : real;
   EtchingRate : Real;
   EtchingRatePiece : Real;
   EtchingBase : Real;
   NumEtch           : Integer;
   I,J               : Integer;

begin
     a := chr(39);
   assign(f,Status.path+TEMPPath+status.filebase+'.LUX');
   reset(f);
   NumLifPts := 0;
```

```
prevmax := -500.0;
prevmin := 500.0;
while not(eof(f)) do begin
  readln(f, t,v1,v2);
  if (v1 > prevmax) then prevmax := v1;
  if (v2 > prevmax) then prevmax := v2;
  if (v1 < prevmin) then prevmin := v1;
  if (v2 < prevmin) then prevmin := v2;
  NumLifPts := NumLifPts + 1;
end;
RunTime := t;
endtime := 10.0*int(t/10.0 +1.0);          { Round it up to next mult. of 10 }
close(f);
GetFileName(filenm,'T',sequence);
assign(f,status.path+PlotPath+filenm+'.pl');
rewrite(f);
writeln(f,'%     Luxtron Temperature Automated Plot Generator');
writeln(f,'      '+status.path+temppath+status.filebase+'.lux');
writeln(f,'%     Timothy J. Dalton (c) 1991');
writeln(f);
writeln(f,'set page format landscape');
writeln(f,'set window x 1.5 to 9.5 y 3 to 7');
writeln(f,'set limits x 0 ',endtime:4);
writeln(f,'set limits y ',prevmin:4,' ',prevmax:4);
writeln(f,'title bottom '+a+'Time (Seconds)'+a);
writeln(f,'title top '+a+status.filebase+a);
writeln(f,'title left '+a+'Temperature (C)'+a);
if (status.lifer = false) then
  writeln(f,'set axis all on')
else
  writeln(f,'set axis all on right off');
if (numlifpts > 1024) then begin
  l := round(numlifpts);
  j := l div 1024;
  writeln(f,'set sample ',(j+1):2);
end;
writeln(f,'insert '+status.path+Temppath+status.filebase+'.lux');
writeln(f,'set intensity 2.5');
writeln(f,'join');
writeln(f,'set intensity default');
writeln(f,'set order x dummy y');
if (numlifpts > 1024) then begin
  l := round(numlifpts);
  j := l div 1024;
  writeln(f,'set sample ',(j+1):2);
end;
writeln(f,'insert '+status.path+temppath+status.filebase+'.lux');
writeln(f,'draw axes');
writeln(f,'join ');

writeln(f,'set legend spacing 1.2 justify left size 1.0');
Assign(f2, Status.Path+InfoPath+Status.FileBase+'.INF');
ReSet(f2);
readln(f2, Dummy);
```

```
    writeln(f,'legend 0.5 2.0 '+a+dummy+a);
    { Read in pre run comments }
    Repeat
      Readln(f2,Dummy);
      writeln(f, 'legend '+a+dummy+a);
      if (SumOpen = True) then writeln(sum,dummy);
    Until (Dummy[1] = '*') and (Dummy[0] > #1);
    { Skip the run Info }
    repeat
      readln(f2,dummy);
    Until (Dummy[1] = '*') and (Dummy[0] > #1);
    { Skip the darss times if there... }
    repeat
      readln(f2,dummy);
    Until (Dummy[1] = '*') and (Dummy[0] > #1);
    writeln(f,' legend 4.0 2.0 '+a+dummy+a);
    repeat
      Readln(f2,Dummy);
      writeln(f, 'legend '+a+dummy+a);
    until eof(f2);
    close(f2);

    if (status.lifer = true) then begin
      writeln(f,'set order x y');
      writeln(f,'set limits x 0 ',endtime:4);
      writeln(f,'set axis all off right on');
      writeln(f,'set labels right on');
      writeln(f,'set sample 1');
      writeln(f,'title right '+a+'Interferometry Signal(v)'+a);
      if (numlifpts > 1024) then begin
        I := round(numlifpts);
        j := I div 1024;
        writeln(f,'set sample ',(j+1):2);
      end;
      writeln(f,'insert '+status.path+liferpath+status.filebase+'.lin');
      writeln(f,'set limits ymin default ymax default');
      writeln(f,'draw axes');
      writeln(f,'join dots');
    end;


   writeln(f,'stamp bottom date time');
   close(f);
end;


PROCEDURE WriteSum;
{ Produce the Initial Plot }
var f2            : text;
    dummy         : string;

    delta  : real;
    prevmax,
    prevmin       : real;
```

```
  EtchTime      : real;
  EtchingRate : Real;
  EtchingRatePiece : Real;
  EtchingBase : Real;
  NumEtch       : Integer;

begin
  If (SumOpen = TRUE) Then Begin
    Assign(f2, Status.Path+InfoPath+Status.FileBase+'.INF');
    ReSet(f2);
    readln(f2, Dummy);
    writeln(Sum);
    writeln(Sum,'FILE: '+Status.FileBase);
    writeln(Sum);
    writeln(Sum);
    writeln(Sum,dummy);

    { Read in pre run comments }
    Repeat
      Readln(f2,Dummy);
      writeln(sum,dummy);
    Until (Dummy[1] = '*') and (Dummy[0] > #1);

    { Skip the run Info }
    repeat
      readln(f2,dummy);
    Until (Dummy[1] = '*') and (Dummy[0] > #1);

    { Skip the darss times if there... }
    repeat
      readln(f2,dummy);
    Until (Dummy[1] = '*') and (Dummy[0] > #1);

    writeln(sum);
    writeln(sum,dummy);

    repeat
      Readln(f2,Dummy);
      writeln(sum,dummy);
    until eof(f2);
    close(f2);

    writeln(sum);
    writeln(sum,'DIAGNOSTICS Used');
    writeln(sum);

    if (Status.Lifer) then writeln(sum,'LIFER');

    if (status.darss) then begin
      writeln(sum,'DARSS');
      writeln(sum,'Grating : ',darssParam.grating:1);
      writeln(sum,'Center  : ',darssParam.center:5:0);
      writeln(sum,'Exposure: ',darssParam.exposure:5);
      writeln(sum,'Correc. : ',darssParam.correction:5:1);
```

```
end;

if (status.comdel) then begin
  writeln(sum,'COMDEL');
end;

if (status.luxtron) then begin
     writeln(sum,'Luxtron Temperature Measurement');
end;

if (status.thr640) then begin
  writeln(sum,'THR640');

  if (status.thr640lda) then begin
    writeln(sum,'LDA');
  end;

  if (status.thr640pmt.WaveScan) then begin
    writeln(sum,'PMT WaveScan');

        if (status.THR640PMT.Detector = 0) then
      writeln(sum,'  Detector Type         = K485')
    else begin
      writeln(sum,'  Detector Type         = K427');
       case status.thr640pmt.gain of
         Gain_1:  writeln(sum,'  A/D Gain Setting       = 1');
         Gain_2:  writeln(sum,'  A/D Gain Setting       = 2');
         Gain_4:  writeln(sum,'  A/D Gain Setting       = 4');
         Gain_8:  writeln(sum,'  A/D Gain Setting       = 8');
      end; {case}
        writeln(sum,'  K427 Gain Setting,    10^',Status.THR640PMT.Gain2:2);
        writeln(sum,'  K427 Rise Time (mSec)  = ',Status.THR640PMT.RiseTime:5);
    end;

    writeln(sum,'  Ending  Wavelength    = ',status.EndingWaveLength:7:1);
        writeln(sum,'  Wavelength Step      = ',status.WaveLengthStep:7:4);
    writeln(sum,'  Grating              = ',status.grating);
    writeln(sum,'  PMT Voltage          = ',status.pmtvoltage);
    writeln(sum,'  No. Pts in Keithley Avg = ',Status.KeitAvg:3);
    writeln(sum,'  Input Slit (microns)   = ',status.inputslit:5:1);
    writeln(sum,'  Output Slit (microns)  = ',status.outputslit:5:1);
  end;

  if (status.thr640pmt.OptEmission) then begin
    writeln(sum,'PMT Optical Emission');

    if (status.THR640PMT.Detector = 0) then
      writeln(sum,'  Detector Type         = K485')
    else begin
      writeln(sum,'  Detector Type         = K427');
       case status.thr640pmt.gain of
         Gain_1:  writeln(sum,'  A/D Gain Setting       = 1');
         Gain_2:  writeln(sum,'  A/D Gain Setting       = 2');
         Gain_4:  writeln(sum,'  A/D Gain Setting       = 4');
```

```
        Gain_8: writeln(sum,'  A/D Gain Setting       = 8');
      end; {case}
        writeln(sum,'  K427 Gain Setting,   10^',Status.THR640PMT.Gain2:2);
        writeln(sum,'  K427 Rise Time (mSec)  = ',Status.THR640PMT.RiseTime:5);
    end;


    writeln(sum,'  Wavelength          = ',status.startingWaveLength:7:1);
    writeln(sum,'  Grating             = ',status.grating);
    writeln(sum,'  PMT Voltage         = ',status.pmtvoltage);
    writeln(sum,'  Input Slit (microns)  = ',status.inputslit:5:1);
    writeln(sum,'  Output Slit (microns)  = ',status.outputslit:5:1);
  end;

end; {Status.THR640}

if (Status.lifer) then begin
  if (NumberExtrema <> 0)  then begin
    EtchTime := 0.0;
    NumEtch := 0;
    prevmax := 0.0;
    theta := theta * 3.141592654 / 180.0;
    EtchingBase := (Lambda/2.0) / sqrt(n2*n2 - n1*n1*sin(theta)*sin(Theta));
    EtchingRate := 0.0;
    prevmin := 0.0;
    writeln(Sum);
    writeln(Sum,'INTERFEROMETRY');
        writeln(Sum);
    writeln(Sum,'  Times   Type Delta  Etching R.');
    writeln(sum);

    for I := 1 to NumberExtrema do begin
      if (ExtremaType[i] = 'Max') then begin
        delta := extrematimes[i] - prevmax;
        prevmax := extrematimes[i];
        end
      else begin
        delta := extrematimes[i] - prevmin;
        prevmin := extrematimes[i];
      end;
      EtchingRatePiece := EtchingBase/delta*60;
      if ((I >= 3) and (I <= NumberExtrema-2)) then begin
        EtchTime := EtchTime + delta;
        EtchingRatePiece := EtchingBase/delta*60;
        EtchingRate  := EtchingRate + EtchingRatePiece;
        NumEtch := NumEtch + 1;
      end;
      writeln(Sum,ExtremaTimes[i]:10:3,' ',
           ExtremaType[i],' ',delta:5:2,' ',
           EtchingRatePiece:10:2);
    end; {For I}

    etchingRate := EtchingRate / NumEtch;
    writeln(Sum);
```

```
        Writeln(Sum,'Lambda = ',lambda:5:1,' nM');
        writeln(Sum,'N2    = ',N2:5:2);
        writeln(Sum,'N1    = ',N1:5:2);
        writeln(Sum,'Theta  = ',theta:6:3,' Deg.');
        writeln(Sum,'ER    = ',EtchingRate:10:2);
        writeln(Sum);

      end; {if NumberExtrema}
    end; {If Status.Lifer}
  end; {If SumOpen = True}
end;


PROCEDURE PlotThR640OptEmm(Sequence:Integer);
{ Plot Optical Emission Data for THR640 }

var f,f2  : text;
   filenm,S        : string;
   a               : char;
   i,j             : Integer;
   Time  : Array[1..2048] of real;
   Inten  : Array[1..2048] of real;
   OptNum          : Integer;

begin
   a := chr(39);
   GetFileName(filenm,'S',sequence);
   assign(f,status.path+PlotPath+filenm+'.pl');
   rewrite(f);
   writeln(f,'%      THR640 PIE Optical Emission Automated Plot Generator');
   writeln(f,'       '+status.path+Optpath+status.filebase+'.OPT');
   writeln(f,'%      Timothy J. Dalton (c) 1991');
   writeln(f);
   writeln(f,'set page format landscape');
   writeln(f,'set window x 1.5 to 9.5 y 3 to 7');
   writeln(f,'title bottom '+a+'Time (Seconds)'+a);
   writeln(f,'title top '+a+status.filebase+a);

   if (DataFilter) then begin
     str(NumFilter:2,s);
     writeln(f,'title left '+a+'PMT Current (nA) [LowPass:'+s+']'+a);
     end
   else
     writeln(f,'title left '+a+'PMT Signal (V)'+a);
     { writeln(f,'title left '+a+'PMT Current (nA)'+a); }

   if (Status.Lifer) then begin
     writeln(f,'set axis all on right off bottom off');
     writeln(f,'set axis left on top on');
     writeln(f,'set limits X 0.0 ',endtime:6:0);
   end;

   if (DataFilter) then begin
     assign(f2,status.path+Optpath+status.filebase+'.OPT');
```

```
    Reset(f2);
    OptNum := 1;
    readln(f2, time[OptNum], inten[OptNum] );
    while ( not(eof(f2)) ) do begin
      OptNum := OptNum + 1;
      readln(f2, time[OptNum], inten[OptNum]);
    end;
    close(f2);

    for i := 1 to NumFilter do
      for J := 2 to OptNum do
        inten[j] := (inten[j] + inten[j-1])/2.0;

    for j := 1 to OptNum do
      writeln(f,time[j]:9:2,inten[j]:10:2);

    end
  else
    writeln(f,'insert '+status.path+Optpath+status.filebase+'.OPT');

  if (status.lifer) then
    writeln(f,'draw axes');
  writeln(f,'join');
  if (status.lifer) then begin
    writeln(f,'set limits X 0.0 ',endtime:6:0);
    writeln(f,'set axis all off right on bottom on');
    writeln(f,'set labels right on');
    writeln(f,'title right '+a+'Interferometry Signal (V)'+a);
    writeln(f,'title bottom '+a+'Time (Seconds)'+a);
    writeln(f,'insert '+status.path+LiferPath+status.filebase+'.LIN');
    writeln(f,'set limits ymin default ymax default');
    writeln(f,'draw axes');
    writeln(f,'join dots');
  end;
  writeln(f,'set legend spacing 1.2 justify left size 1.1 font courier');
  writeln(f,'legend 0.5 2.0 '+a+'THR-640 Wave Scan'+a);
  writeln(f,'legend '+a+'Grating                    : '+Status.grating+a);
  str(status.PMTVoltage:6:0, S);
  writeln(f,'legend '+a+'PMT Voltage                : '+s+a);
  str(Status.StartingWaveLength:7:1, S);
  writeln(f,'legend '+a+'Wavelength                 : '+S+a);

  if (status.THR640PMT.Detector = 0) then
    writeln(f,'legend '+a+'Detector Type              : K485'+a)
  else begin
    writeln(f,'legend '+a+'Detector Type              : K427'+a);
    case status.thr640pmt.gain of
      Gain_1:writeln(f,'legend '+a+'A/D Gain Setting          : 1'+a);
      Gain_2:writeln(f,'legend '+a+'A/D Gain Setting          : 2'+a);
      Gain_4:writeln(f,'legend '+a+'A/D Gain Setting          : 4'+a);
      Gain_8:writeln(f,'legend '+a+'A/D Gain Setting          : 8'+a);
    end; {case}
    str(Status.THR640PMT.Gain2:2, S);
    writeln(f,'legend '+a+'K427 Gain Setting, 10^x    : '+S+a);
```

```
    str(Status.THR640PMT.RiseTime:5, S);
    writeln(f,'legend '+a+'K427 Rise Time Setting (mSec): '+S+a);
  end;

  str(Status.InputSlit:5:1, S);
  writeln(f,'legend '+a+'Input Slit             : '+S+' mm'+a);
  Writeln(f,'case '+a+'                        g '+a);
  str(Status.OutputSlit:5:1, S);
  Writeln(f,'legend '+a+'Output Slit            : '+S+' mm'+a);
  writeln(f,'case '+a+'                        g '+a);
  writeln(f,'set symbol 0N');
  writeln(f,'0.0 0.0');
  writeln(f,'plot');
  writeln(f,'stamp bottom date time');
  close(f);
end;


PROCEDURE PlotThR640Wave(Sequence:Integer);

{ Plot Wave Scan Data for THR640 }

var f          : text;
  filenm,S     : string;
  a            : char;

begin
  a := chr(39);
  GetFileName(filenm,'S',sequence);
  assign(f,status.path+PlotPath+filenm+'.pl');
  rewrite(f);
  writeln(f,'%     THR640 PIE Wave Scan Automated Plot Generator');
  writeln(f,'      '+status.path+Optpath+status.filebase+'.OPT');
  writeln(f,'%     Timothy J. Dalton (c) 1991');
  writeln(f);
  writeln(f,'set page format landscape');
  writeln(f,'set window x 1.5 to 9.5 y 3 to 7');
  writeln(f,'title bottom '+a+'Wavelength (Ang.) '+a);
  writeln(f,'title top '+a+status.filebase+a);
  writeln(f,'title left '+a+'PMT Current (nA)'+a);
  writeln(f,'set order x y');
  writeln(f,'set symbol 2N');
  writeln(f,'insert '+status.path+Optpath+status.filebase+'.OPT');
  writeln(f,'join');
  writeln(f,'plot');
  writeln(f,'set legend spacing 1.2 justify left size 1.1 font courier');
  writeln(f,'legend 0.5 2.0 '+a+'THR-640 Wave Scan'+a);
  writeln(f,'legend '+a+'Grating                : '+Status.grating+a);
  str(status.PMTVoltage:6:0, S);
  writeln(f,'legend '+a+'PMT Voltage            : '+s+a);
  str(Status.StartingWaveLength:7:1, S);
  writeln(f,'legend '+a+'Starting Wavelength    : '+S+a);
  str(status.EndingWaveLength:7:1, S);
  writeln(f,'legend '+a+'Ending Wavelength      : '+S+a);
```

```
   str(Status.WaveLengthStep:7:4, S);
   writeln(f,'legend '+a+'Wavelength Step            : '+S+a);

   if (status.THR640PMT.Detector = 0) then begin
      writeln(f,'legend '+a+'Detector Type            : K485'+a);
      str(Status.KeitAvg:3, S);
      writeln(f,'legend '+a+'No. Readings in Keithley Avg : '+S+a);
      end
   else begin
      writeln(f,'legend '+a+'Detector Type            : K427'+a);
      case status.thr640pmt.gain of
        Gain_1:writeln(f,'legend '+a+'A/D Gain Setting          : 1'+a);
        Gain_2:writeln(f,'legend '+a+'A/D Gain Setting          : 2'+a);
        Gain_4:writeln(f,'legend '+a+'A/D Gain Setting          : 4'+a);
        Gain_8:writeln(f,'legend '+a+'A/D Gain Setting          : 8'+a);
      end; {case}
      str(Status.THR640PMT.Gain2:2, S);
      writeln(f,'legend '+a+'K427 Gain Setting, 10^x     : '+S+a);
      str(Status.THR640PMT.RiseTime:5, S);
      writeln(f,'legend '+a+'K427 Rise Time Setting (mSec): '+S+a);

   end;

   str(Status.InputSlit:5:1, S);
   writeln(f,'legend '+a+'Input Slit                : '+S+' mm'+a);
   Writeln(f,'case '+a+'                         g '+a);
   str(Status.OutputSlit:5:1, S);
   Writeln(f,'legend '+a+'Output Slit                : '+S+' mm'+a);
   writeln(f,'case '+a+'                         g '+a);
   writeln(f,'stamp bottom date time');
   close(f);
end;


PROCEDURE PlotThisSpectra(Sequence:Integer);
{ Produce a Spectra Plot}
VAR
   f              : text;
   filenm         : string;
   a              : char;
   LineNo,
   Column,I       : Integer;
   done  : boolean;

begin

writeln('IN PLOT THIS SPECTRUM!!!');

   a := chr(39);
   GetFileName(filenm,'S',sequence);
   assign(f,status.path+PlotPath+filenm+'.pl');
   rewrite(f);
   writeln(f,'%     DARSS Spectra Automated Plot Generator');
   writeln(f,'      '+status.path+optpath+fname+'.opt :',NumberSpectra:3);
```

```
  writeln(f,'%      Timothy J. Dalton (c) 1991');
  writeln(f);
  writeln(f,'set page format landscape');
  writeln(f,'set window x 1.5 to 9.5 y 3 to 7');
  writeln(f,'set limits ymin 0');
  writeln(f,'title bottom '+a+'Wavelength (A)'+a);
  writeln(f,'case '+a+'         M '+a);
  writeln(f,'title top '+a+fname+':',NumberSpectra:3,
       ' (',TimeG:7:2,' sec)'+a);
  writeln(f,'title left '+a+'Counts'+a);
  for I := 5 to 1020 do
    Writeln(f,DARSSParam.Wave[i]:10:4, ' ',DARSSParam.PData[i]:5,' ');
  writeln(f,'join');
  writeln(f,'set legend spacing 1.2 justify left size 0.85 font courier');
  writeln(f,'legend 0.5 2.0 '+a+'Peak Assignments'+a);
  writeln(f,'legend '+a+'Spectra Number: ',NumberSpectra:3,'  Time = ',
        TimeG:9:4,a);
  writeln(f,'legend '+a+' #   Start    Stop     Max      Area'+a);
  LineNo := 3;  Column := 0;
  if (actin) then begin
    for I := 1 to NumberPeaks do begin
      if (i <=42) then begin
        LineNo := LineNo + 1;
        if ((LineNo Mod 17) = 0) then
          write(f,'legend ',(0.5+2.5*int(LineNo/16)):3:1,' 2.0 ')
        else
          write(f,'legend ');
        with DARSSParam do
          writeln(f,a,i:2,' ',Wave[PStart[i]]:10:4,' ',
          Wave[PEnd[i]]:10:4, ' ', Wave[PMax[i]]:10:4, ' ',
          PArea[i]*10*DARSSParam.DataMax:10:4,a);
      end; {If}
    end; {For}
  end; {if actin}
  writeln(f,'legend 9.6, 7, '+a+'DARSS'+a);
  writeln(f,'legend '+a+'Grating : ',darssParam.grating:1,a);
  writeln(f,'legend '+a+'Center  : ',darssParam.center:5:0,a);
  writeln(f,'legend '+a+'Exposure: ',darssParam.exposure:5,a);
  writeln(f,'legend '+a+'Correc. : ',darssParam.correction:5:1,a);
  if (actin = true) then begin
    writeln(f,'legend 8.0 2.0 '+a+'Actinometry Parameters'+a);
    writeln(f,'legend '+a+'Peak ID  Peak  Ss   Act Ratio'+a);
    writeln(f,'case  '+a+'         SG         '+a);
    Act[NumPeak].ratio := 1.0;
    for I := 1 to NumPeak do
      writeln(f,'legend '+a,act[i].GasName:7,' ',act[i].peak:5:0,
          act[i].delta:4:0,'  ',act[i].ratio:8:4,a);
  end;
  writeln(f,'stamp bottom date time');
  close(f);
end;


PROCEDURE PlotActin(Sequence:Integer);
```

```
{ Produce a Plot of Actinometry Data Plot}
VAR
  f              : text;
  filenm         : string;
  a              : char;
  LineNo,
  Column,I,J     : Integer;
  done  : boolean;
  Symbols        : Array[1..10] of string;

begin
  Symbols[1] := '2N';
  Symbols[2] := '3N';
  Symbols[3] := '5N';
  Symbols[4] := '9N';
  Symbols[5] := '6N';
  Symbols[6] := '2P';
  Symbols[7] := '3P';
  Symbols[8] := '5P';
  Symbols[9] := '9P';
  Symbols[10]:= '6P';

  a := chr(39);
  GetFileName(filenm,'A',sequence);
  assign(f,status.path+PlotPath+filenm+'.pl');
  rewrite(f);
  writeln(f,'%      Plasma Induced Emission Actinometry Automated Plot Generator');
  writeln(f,'       '+status.path+optpath+fname+'.dat :',NumberSpectra:3);
  writeln(f,'%      Timothy J. Dalton (c) 1991');
  writeln(f);
  writeln(f,'set page format landscape');
  writeln(f,'set window x 1.5 to 9.5 y 3 to 7');
  writeln(f,'set limits x 0 ',endtime:4);
  writeln(f,'set limits y ',(0.9*actymin):10:2,' ',(1.1*actymax):10:2);
  writeln(f,'title bottom '+a+'Time (Seconds)'+a);
  writeln(f,'title top '+a+fname+a);
  writeln(f,'title left '+a+'PIE Actinometric Ratios'+a);
  for I := 1 to NumPeak -1 do begin
    write(f,'set order x');
    for J := 1 to I-1 do
      write(f,' dummy');
    writeln(f,' y');
    writeln(f,'set symbol '+symbols[i]);
    writeln(f,'insert '+status.path+optpath+fname+'.dat');
    writeln(f,'join');
    writeln(f,'plot');
  end;
  writeln(f,'set legend spacing 1.2 justify left size 1.5');
  for I := 1 to NumPeak-1 do begin
   write(f,'legend');
   if (I=1) then write(f,' 1 2');
   writeln(f,' symbol '+symbols[i]+' '+a+'O ',act[i].gasname:3,
         ' ',act[i].peak:4:0,'/',act[NumPeak].GasName:2,
         ' ',act[NumPeak].Peak:4:0,a);
```

```
    writeln(f,' case '+a+'L              '+a);
  end;
  writeln(f,'stamp bottom date time');
  close(f);
end;

PROCEDURE PlotRAW(Sequence:Integer);
{ Produce a Plot of Peak Areas Data Plot}
VAR
  f              : text;
  filenm         : string;
  a              : char;
  LineNo,
  Column,I,J     : Integer;
  done  : boolean;
  Symbols        : Array[1..10] of string;

begin
  Symbols[1] := '2N';
  Symbols[2] := '3N';
  Symbols[3] := '5N';
  Symbols[4] := '9N';
  Symbols[5] := '6N';
  Symbols[6] := '2P';
  Symbols[7] := '3P';
  Symbols[8] := '5P';
  Symbols[9] := '9P';
  Symbols[10]:= '6P';

  a := chr(39);
  GetFileName(filenm,'R',sequence);
  assign(f,status.path+PlotPath+filenm+'.pl');
  rewrite(f);
  writeln(f,'%     Plasma Induced Emission Peak Area Automated Plot Generator');
  writeln(f,'      '+status.path+optpath+fname+'.dat :',NumberSpectra:3);
  writeln(f,'%     Timothy J. Dalton (c) 1991');
  writeln(f);
  writeln(f,'set page format landscape');
  writeln(f,'set window x 1.5 to 9.5 y 3 to 7');
  writeln(f,'set limits x 0 ',endtime:4);
  writeln(f,'set limits y ',(0.9*rawymin):10:2,' ',(1.1*rawymax):10:2);
  writeln(f,'title bottom '+a+'Time (Seconds)'+a);
  writeln(f,'title top '+a+fname+a);
  writeln(f,'title left '+a+'PIE Peak Area'+a);
  for I := 1 to NumPeak do begin
    write(f,'set order x ');
    {Skip Actin Data}
    for J := 1 to NumPeak -1 do
      write(f,' dummy');
    for J := ' to I-1 do
      write(f,' dummy');
    writeln(f,' y');
    writeln(f,'set symbol '+symbols[i]);
    writeln(f,'insert '+status.path+optpath+fname+'.dat');
```

```
    writeln(f,'join');
    writeln(f,'plot');
  end;
  writeln(f,'set legend spacing 1.2 justify left size 1.5');
  for I := 1 to NumPeak do begin
   write(f,'legend');
   if (I=1) then write(f,' 1 2');
   writeln(f,' symbol '+symbols[i]+' '+a+'O ',act[i].gasname:3,
        ' ',act[i].peak:4:0,a);
   writeln(f,' case '+a+'L       '+a);
  end;
  writeln(f,'stamp bottom date time');
  close(f);
end;


begin
end.
```

```
unit plot2;

{ Generate ProPlot Files }

INTERFACE
Uses Dos, Crt, Vars, Misc, MSOPTVG, plot;

PROCEDURE PlotElec1(Sequence:Integer);
PROCEDURE PlotElec2(Sequence:Integer);
PROCEDURE PlotElec1OLD(Sequence:Integer);
PROCEDURE PlotElec2OLD(Sequence:Integer);

IMPLEMENTATION

PROCEDURE PlotElec1OLD(Sequence:Integer);
{ Produce a Plot of Electrical Characteristics: I, V, & P}
var
 f,f2     : Text;
 Num      : Real;
 Freq  : Integer;
 filenm: string;
 i,j      : integer;
 a        : char;
 dummy          : string;

 IAvg,
 VAvg,
 PAvg   : Real;
 NumAvg,
 nn       : integer;
 pow, volt, cur, imped, phase, dcb : real;


begin
 IAvg := 0.0; VAvg := 0.0; PAvg := 0.0;
 NumAvg := 0;
 Assign(f,status.path+IVPath+status.filebase+'.iv');
 reset(f);
 for I:=1 to 7 do readln(f,dummy);
 while not(eof(f)) do
   readln(f, Num);
 close(f);
 freq := round(Num/endtime);

 { Determine the Sampling Frequency on the Comdel }
 if (freq <= 1.5) then
   freq := 1
 else if (freq <= 2.5 ) then
   freq := 2
 else if (freq <= 5.5) then
   freq := 5
 else
   freq := 10;
```

```
assign(f2,status.path+ivpath+status.filebase+'.iv');
reset(f2);
for I := 1 to 7 do
  readln(f2);

readln(f2, nn, pow, volt, cur, imped, phase, dcb);

while (not(eof(f2)) and (nn <= 0.60*freq*runtime)) do begin
  if (nn > 0.40*freq*runtime) then begin
    numavg := numavg + 1;
    IAvg := IAvg + cur;
    VAvg := VAvg + volt;
    PAvg := PAvg + pow;
  end;
  readln(f2, nn, pow, volt, cur, imped, phase, dcb);
end;
IAvg := IAvg / numavg;
VAvg := VAvg / numavg;
PAvg := PAvg / numavg;

close(f2);

if (SumOpen = True) then begin
  writeln(sum);
  writeln(sum,'COMDEL Electrical Characterization');
  writeln(sum);
  writeln(Sum,'Average Time is from ',0.40*runtime:6:1,' to ',
       0.60*runtime:6:1,' sec.');
  writeln(sum);
  writeln(sum,'POWER      = ',PAvg:10:4);
  writeln(sum,'VOLTAGE    = ',VAvg:10:4);
  writeln(sum,'CURRENT    = ',IAvg:10:4);
end;

a := chr(39);
GetFileName(filenm,'I',sequence);
assign(f,status.path+PlotPath+filenm+'.pl');
rewrite(f);
writeln(f,'%     Electrical Data Plot Generator 1');
writeln(f,'     '+status.path+ivpath+status.filebase+'.iv');
writeln(f,'%     Timothy J. Dalton (c) 1991');
writeln(f);
writeln(f,'set page format portrait');

writeln(f,'set window x 1.0 to 7.5 y 1.0 to 3.333');
writeln(f,'title left '+a+'rf Current (Amps)'+a);
writeln(f,'set order x dummy dummy y');
writeln(f,'set axis all on right off bottom off');
writeln(f,'set axis left on top on');
writeln(f,'set limits x 0 ',(endtime*freq):6);
if (num > 1024) then begin
  I := round(num);
  j := I div 1024;
  writeln(f,'set sample ',(j+1):2);
```

```
end;
writeln(f,'insert '+status.path+ivpath+status.filebase+'.iv');
writeln(f,'draw axes');
writeln(f,'join');
writeln(f,'set order x y');
writeln(f,'set limits x 0 ',endtime:4);
writeln(f,'set axis all off right on bottom on');
writeln(f,'set labels right on');
writeln(f,'set sample 1');
writeln(f,'title right '+a+'Interferometry Signal(v)'+a);
writeln(f,'title bottom '+a+'Time (Sec)   ',
        'Avg[',0.40*runtime:6:1,',',0.60*runtime:6:1,'sec] = ',
         IAvg:10:4,a);
writeln(f,'insert '+status.path+liferpath+status.filebase+'.lin');
writeln(f,'set limits ymin default ymax default');
writeln(f,'draw axes');
writeln(f,'join dots');

writeln(f,'set window x 1.0 to 7.5 y 4.333 to 6.666');
writeln(f,'title left '+a+'rf Voltage (Volts)'+a);
writeln(f,'set order x dummy y');
writeln(f,'set axis all on right off bottom off');
writeln(f,'set axis left on top on');
writeln(f,'set limits x 0 ',(endtime*freq):6);
if (num > 1024) then begin
  I := round(num);
  j := I div 1024;
  writeln(f,'set sample ',(j+1):2);
end;
writeln(f,'insert '+status.path+ivpath+status.filebase+'.iv');
writeln(f,'draw axes');
writeln(f,'join');
writeln(f,'set order x y');
writeln(f,'set limits x 0 ',endtime:4);
writeln(f,'set axis all off right on bottom on');
writeln(f,'set labels right on');
writeln(f,'set sample 1');
writeln(f,'title right '+a+'Interferometry Signal(v)'+a);
writeln(f,'title bottom '+a+'Time (Sec)   ',
        'Avg[',0.40*runtime:6:1,',',0.60*runtime:6:1,'sec] = ',
         VAvg:10:4,a);
writeln(f,'insert '+status.path+liferpath+status.filebase+'.lin');
writeln(f,'set limits ymin default ymax default');
writeln(f,'draw axes');
writeln(f,'join dots');

writeln(f,'set window x 1.0 to 7.5 y 7.666 to 10.000');
writeln(f,'title left '+a+'Actual Power (Watts)'+a);
writeln(f,'title top '+a+status.filebase+a);
writeln(f,'set order x y');
writeln(f,'set axis all on right off bottom off');
writeln(f,'set axis left on top on');
writeln(f,'set limits x 0 ',(endtime*freq):6);
if (num > 1024) then begin
```

```
      I := round(num);
      j := I div 1024;
      writeln(f,'set sample ',(j+1):2);
    end;
    writeln(f,'insert '+status.path+ivpath+status.filebase+'.iv');
    writeln(f,'draw axes');
    writeln(f,'join');
    writeln(f,'set order x y');
    writeln(f,'set limits x 0 ',endtime:4);
    writeln(f,'set axis all off right on bottom on');
    writeln(f,'set labels right on');
    writeln(f,'set sample 1');
    writeln(f,'title right '+a+'Interferometry Signal(v)'+a);
    writeln(f,'title bottom '+a+'Time (Sec)   ',
           'Avg[',0.40*runtime:6:1,',',0.60*runtime:6.1,'sec] = ',
            PAvg:10:4,a);
    writeln(f,'insert '+status.path+liferpath+status.filebase+'.lin');
    writeln(f,'set limits ymin default ymax default');
    writeln(f,'draw axes');
    writeln(f,'join dots');

    writeln(f,'stamp bottom date time');
    close(f);
end;


PROCEDURE PlotElec2OLD(Sequence:Integer);
{ Produce a Plot of Electrical Characteristics: I, V, & P}
var
 f,f2      : Text;
 Num    : Real;
 Freq  : Integer;
 filenm: string;
 i,j       : integer;
 a         : char;
 dummy              : string;

 ZAvg,
 TAvg,
 DAvg  : Real;
 NumAvg,
 nn       : integer;
 pow, volt, cur, imped, phase, dcb : real;

begin
 ZAvg := 0.0; TAvg := 0.0; DAvg := 0.0;
 NumAvg := 0;
 Assign(f,status.path+IVPath+status.filebase+'.iv');
 reset(f);
 for I:=1 to 7 do readln(f,dummy);
 while not(eof(f)) do
  readln(f, Num);
 close(f);
 freq := round(Num/endtime);
```

```
{ Determine the Sampling Frequency on the Comdel }
if (freq <= 1.5) then
  freq := 1
else if (freq <= 2.5 ) then
  freq := 2
else if (freq <= 5.5) then
  freq := 5
else
  freq := 10;

  assign(f2,status.path+ivpath+status.filebase+'.iv');
  reset(f2);
  for I := 1 to 7 do
    readln(f2);

  readln(f2, nn, pow, volt, cur, imped, phase, dcb);

  while (not(eof(f2)) and (nn <= 0.60*freq*runtime)) do begin
    if (nn > 0.40*freq*runtime) then begin
      numavg := numavg + 1;
      ZAvg := ZAvg + imped;
      TAvg := TAvg + phase;
      DAvg := DAvg + dcb;
    end;
    readln(f2, nn, pow, volt, cur, imped, phase, dcb);
  end;
  ZAvg := ZAvg / numavg;
  TAvg := TAvg / numavg;
  DAvg := DAvg / numavg;

  close(f2);

  if (SumOpen = True) then begin
    writeln(sum,'IMPEDANCE   = ',ZAvg:10:4);
    writeln(sum,'PHASE ANGLE = ',TAvg:10:4);
    writeln(sum,'DC BIAS     = ',DAvg:10:4);
    writeln(sum)
  end;

  a := chr(39);
  GetFileName(filenm,'Z',sequence);
  assign(f,status.path+PlotPath+filenm+'.pl');
  rewrite(f);
  writeln(f,'%      Electrical Data Plot Generator 2');
  writeln(f,'       '+status.path+ivpath+status.filebase+'.iv');
  writeln(f,'%      Timothy J. Dalton (c) 1991');
  writeln(f);
  writeln(f,'set page format portrait');

  writeln(f,'set window x 1.0 to 7.5 y 1.0 to 3.333');
  writeln(f,'title left '+a+'DC Bias (Volts)'+a);
  writeln(f,'set order x dummy dummy dummy dummy dummy y');
  writeln(f,'set axis all on right off bottom off');
  writeln(f,'set axis left on top on');
```

```
writeln(f,'set limits x 0 ',(endtime*freq):6);
if (num > 1024) then begin
  I := round(num);
  j := I div 1024;
  writeln(f,'set sample ',(j+1):2);
end;
writeln(f,'insert '+status.path+ivpath+status.filebase+'.iv');
writeln(f,'draw axes');
writeln(f,'join');
writeln(f,'set order x y');
writeln(f,'set limits x 0 ',endtime:4);
writeln(f,'set axis all off right on bottom on');
writeln(f,'set labels right on');
writeln(f,'set sample 1');
writeln(f,'title right '+a+'Interferometry Signal(v)'+a);
writeln(f,'title bottom '+a+'Time (Sec)   ',
       'Avg[',0.40*runtime:6:1,',',0.60*runtime:6:1,'sec] = ',
        DAvg:10:4,a);
writeln(f,'insert '+status.path+liferpath+status.filebase+'.lin');
writeln(f,'set limits ymin default ymax default');
writeln(f,'draw axes');
writeln(f,'join dots');

writeln(f,'set window x 1.0 to 7.5 y 4.333 to 6.666');
writeln(f,'title left '+a+'Phase Angle (Deg.)'+a);
writeln(f,'set order x dummy dummy dummy dummy y');
writeln(f,'set axis all on right off bottom off');
writeln(f,'set axis left on top on');
writeln(f,'set limits x 0 ',(endtime*freq):6);
writeln(f,'set limits y -90 -80 ');
if (num > 1024) then begin
  I := round(num);
  j := I div 1024;
  writeln(f,'set sample ',(j+1):2);
end;
writeln(f,'insert '+status.path+ivpath+status.filebase+'.iv');
writeln(f,'draw axes');
writeln(f,'join');
writeln(f,'set order x y');
writeln(f,'set limits x 0 ',endtime:4);
writeln(f,'set axis all off right on bottom on');
writeln(f,'set labels right on');
writeln(f,'set sample 1');
writeln(f,'title right '+a+'Interferometry Signal(v)'+a);
writeln(f,'title bottom '+a+'Time (Sec)   ',
       'Avg[',0.40*runtime:6:1,',',0.60*runtime:6:1,'sec] = ',
        TAvg:10:4,a);
writeln(f,'insert '+status.path+liferpath+status.filebase+'.lin');
writeln(f,'set limits ymin default ymax default');
writeln(f,'draw axes');
writeln(f,'join dots');

writeln(f,'set window x 1.0 to 7.5 y 7.666 to 10.000');
writeln(f,'title left '+a+'Plasma Impedance (Ohms)'+a);
```

```
  writeln(f,'title top '+a+status.filebase+a);
  writeln(f,'set order x dummy dummy dummy y');
  writeln(f,'set axis all on right off bottom off');
  writeln(f,'set axis left on top on');
  writeln(f,'set limits x 0 ',(endtime*freq):6);
  writeln(f,'set limits y 0 50');
  if (num > 1024) then begin
    I := round(num);
    j := I div 1024;
    writeln(f,'set sample ',(j+1):2);
  end;
  writeln(f,'insert '+status.path+ivpath+status.filebase+'.iv');
  writeln(f,'draw axes');
  writeln(f,'join');
  writeln(f,'set order x y');
  writeln(f,'set limits x 0 ',endtime:4);
  writeln(f,'set axis all off right on bottom on');
  writeln(f,'set labels right on');
  writeln(f,'set sample 1');
  writeln(f,'title right '+a+'Interferometry Signal(v)'+a);
  writeln(f,'title bottom '+a+'Time (Sec)   ',
         'Avg[',0.40*runtime:6:1,',',0.60*runtime:6:1,'sec] = ',
          ZAvg:10:4,a);
  writeln(f,'insert '+status.path+liferpath+status.filebase+'.lin');
  writeln(f,'set limits ymin default ymax default');
  writeln(f,'draw axes');
  writeln(f,'join dots');

  writeln(f,'stamp bottom date time');
  close(f);
end;

PROCEDURE PlotElec1(Sequence:Integer);
{ Produce a Plot of Electrical Characteristics: I, V, & P}
var
 f,f2     : Text;
 Num    : Real;
 Freq  : Integer;
 filenm: string;
 i,j       : integer;
 a         : char;
 dummy             : string;

 IAvg,
 VAvg,
 PAvg   : Real;
 STime  : Real;    {Sample Time}
 NumAvg,
 nn, numpts        : integer;
 pow, volt, cur, imped, phase, dcb : real;


 begin
  IAvg := 0.0; VAvg := 0.0; PAvg := 0.0;
```

```
NumAvg := 0; NumPts := 0;
Assign(f,status.path+IVPath+status.filebase+'.iv');
reset(f);
for I:=1 to 3 do readln(f,dummy);
while not(eof(f)) do begin
  readln(f, Num);
  numpts := numpts + 1;
end;
close(f);
if (Status.lifer = false) then begin
  RunTime := Num;
  endtime := 10.0*int(Num/10.0 +1.0);      { Round it up to next mult. of 10 }
end;

assign(f2,status.path+ivpath+status.filebase+'.iv');
reset(f2);
for I := 1 to 3 do
    readln(f2);

 readln(f2, STime, pow, volt, cur, imped, phase, dcb);

 while (not(eof(f2)) and (STime <= 0.60*Runtime)) do begin
   if (STime > 0.40*RunTime) then begin
     numavg := numavg + 1;
     IAvg := IAvg + cur;
     VAvg := VAvg + volt;
     PAvg := PAvg + pow;
   end;
   readln(f2, STime, pow, volt, cur, imped, phase, dcb);
 end;
 IAvg := IAvg / numavg;
 VAvg := VAvg / numavg;
 PAvg := PAvg / numavg;

 close(f2);

 if (SumOpen = True) then begin
   writeln(sum);
   writeln(sum,'COMDEL Electrical Characterization');
   writeln(sum);
   writeln(Sum,'Average Time is from ',0.40*Runtime:6:1,' to ',
        0.60*Runtime:6:1,' sec.');
   writeln(sum);
   writeln(sum,'POWER      = ',PAvg:10:4);
   writeln(sum,'VOLTAGE    = ',VAvg:10:4);
   writeln(sum,'CURRENT    = ',IAvg:10:4);
end;

a := chr(39);
GetFileName(filenm,'I',sequence);
assign(f,status.path+PlotPath+filenm+'.pl');
rewrite(f);
writeln(f,'%      Electrical Data Plot Generator 1');
writeln(f,'       '+status.path+ivpath+status.filebase+'.iv');
```

```
writeln(f,'%      Timothy J. Dalton (c) 1991');
writeln(f);
writeln(f,'set page format portrait');

writeln(f,'set window x 1.0 to 7.5 y 1.0 to 3.333');
writeln(f,'title left '+a+'rf Current (Amps)'+a);
writeln(f,'set order x dummy dummy y');
if (status.lifer = false) then
  writeln(f,'set axis all on')
else
  writeln(f,'set axis all on right off');
writeln(f,'set axis left on top on');
writeln(f,'set limits x 0 ',endtime:6);

if (numpts > 1024) then begin
  I := round(numpts);
  j := I div 1024;
  writeln(f,'set sample ',(j+1):2);
end;
writeln(f,'insert '+status.path+ivpath+status.filebase+'.iv');
if (status.lifer = false) then
  writeln(f,'title bottom '+a+'Time (Sec)  ',
  'Avg[',0.40*runtime:6:1,',',0.60*runtime:6:1,'sec] = ',
  IAvg:10:4,a);


writeln(f,'draw axes');
writeln(f,'join');
if (status.lifer = true) then begin
  writeln(f,'set order x y');
  writeln(f,'set limits x 0 ',endtime:4);
  writeln(f,'set axis all off right on');
  writeln(f,'set labels right on');
  writeln(f,'set sample 1');
  writeln(f,'title right '+a+'Interferometry Signal(v)'+a);
  writeln(f,'title bottom '+a+'Time (Sec)  ',
        'Avg[',0.40*runtime:6:1,',',0.60*runtime:6:1,'sec] = ',
         IAvg:10:4,a);
  if (numlifpts > 1024) then begin
    I := round(numlifpts);
    j := I div 1024;
    writeln(f,'set sample ',(j+1):2);
  end;
  writeln(f,'insert '+status.path+liferpath+status.filebase+'.lin');
  writeln(f,'set limits ymin default ymax default');
  writeln(f,'draw axes');
  writeln(f,'join dots');
end;

writeln(f,'set window x 1.0 to 7.5 y 4.333 to 6.666');
writeln(f,'title left '+a+'rf Voltage (Volts)'+a);
writeln(f,'set order x dummy y');
if (status.lifer = false) then
  writeln(f,'set axis all on')
```

```
else
  writeln(f,'set axis all on right off');
writeln(f,'set axis left on top on');
writeln(f,'set limits x 0 ',endtime:6);

  if (numpts > 1024) then begin
  I := round(numpts);
  j := I div 1024;
  writeln(f,'set sample ',(j+1):2);
end;
writeln(f,'insert '+status.path+ivpath+status.filebase+'.iv');
if (status.lifer = false) then
  writeln(f,'title bottom '+a+'Time (Sec)  ',
  'Avg[',0.40*runtime:6:1,',',0.60*runtime:6:1,'sec] = ',
  VAvg:10:4,a);

writeln(f,'draw axes');
writeln(f,'join');
if (Status.lifer = true) then begin
  writeln(f,'set order x y');
  writeln(f,'set limits x 0 ',endtime:4);
  writeln(f,'set axis all off right on bottom on');
  writeln(f,'set labels right on');
  writeln(f,'set sample 1');
  writeln(f,'title right '+a+'Interferometry Signal(v)'+a);
  writeln(f,'title bottom '+a+'Time (Sec)  ',
        'Avg[',0.40*runtime:6:1,',',0.60*runtime:6:1,'sec] = ',
         VAvg:10:4,a);
  if (numlifpts > 1024) then begin
    I := round(numlifpts);
    j := I div 1024;
    writeln(f,'set sample ',(j+1):2);
  end;

  writeln(f,'insert '+status.path+liferpath+status.filebase+'.lin');
  writeln(f,'set limits ymin default ymax default');
  writeln(f,'draw axes');
  writeln(f,'join dots');
end;

writeln(f,'set window x 1.0 to 7.5 y 7.666 to 10.000');
writeln(f,'title left '+a+'Actual Power (Watts)'+a);
writeln(f,'title top '+a+status.filebase+a);
writeln(f,'set order x y');
if (status.lifer = false) then
  writeln(f,'set axis all on')
else
  writeln(f,'set axis all on right off');
writeln(f,'set axis left on top on');
writeln(f,'set limits x 0 ',endtime:6);

  if (numpts > 1024) then begin
  I := round(numpts);
  j := I div 1024;
```

```
    writeln(f,'set sample ',(j+1):2);
  end;
  writeln(f,'insert '+status.path+ivpath+status.filebase+'.iv');
  if (status.lifer = false) then
    writeln(f,'title bottom '+a+'Time (Sec)   ',
    'Avg[',0.40*runtime:6:1,',',0.60*runtime:6:1,'sec] = ',
    PAvg:10:4,a);

  writeln(f,'draw axes');
  writeln(f,'join');
  if (Status.lifer = true) then begin
    writeln(f,'set order x y');
    writeln(f,'set limits x 0 ',endtime:4);
    writeln(f,'set axis all off right on');
    writeln(f,'set labels right on');
    writeln(f,'set sample 1');
    writeln(f,'title right '+a+'Interferometry Signal(v)'+a);
    writeln(f,'title bottom '+a+'Time (Sec)   ',
          'Avg[',0.40*runtime:6:1,',',0.60*runtime:6:1,'sec] = ',
           PAvg:10:4,a);
    if (numlifpts > 1024) then begin
      I := round(numlifpts);
      j := I div 1024;
      writeln(f,'set sample ',(j+1):2);
    end;

    writeln(f,'insert '+status.path+liferpath+status.filebase+'.lin');
    writeln(f,'set limits ymin default ymax default');
    writeln(f,'draw axes');
    writeln(f,'join dots');
  end;

  writeln(f,'stamp bottom date time');
  close(f);
end;


PROCEDURE PlotElec2(Sequence:Integer);
{ Produce a Plot of Electrical Characteristics: I, V, & P}
var
 f,f2     : Text;
 Num     : Real;
 Freq  : Integer;
 filenm: string;
 i,j       : integer;
 a         : char;
 dummy             : string;

 ZAvg,
 TAvg,
 DAvg  : Real;
 STime : Real;
 NumAvg,
 nn, numpts        : integer;
```

```
 pow, volt, cur, imped, phase, dcb : real;

begin
 ZAvg := 0.0; TAvg := 0.0; DAvg := 0.0;
 NumAvg := 0; NumPts := 0;
 Assign(f,status.path+IVPath+status.filebase+'.iv');
 reset(f);
 for I:=1 to 3 do readln(f,dummy);
 while not(eof(f)) do begin
  readln(f, Num);
  numpts := numpts + 1;
 end;
 close(f);
 if (Status.lifer = false) then begin
  RunTime := Num;
  endtime := 10.0*int(Num/10.0 +1.0);       { Round it up to next mult. of 10 }
 end;

  assign(f2,status.path+ivpath+status.filebase+'.iv');
  reset(f2);
  for I := 1 to 3 do
    readln(f2);

  readln(f2, STime, pow, volt, cur, imped, phase, dcb);

  while (not(eof(f2)) and (STime <= 0.60*runtime)) do begin
    if (STime > 0.40*runtime) then begin
      numavg := numavg + 1;
      ZAvg := ZAvg + imped;
      TAvg := TAvg + phase;
      DAvg := DAvg + dcb;
    end;
    readln(f2, STime, pow, volt, cur, imped, phase, dcb);
  end;
  ZAvg := ZAvg / numavg;
  TAvg := TAvg / numavg;
  DAvg := DAvg / numavg;

  close(f2);

  if (SumOpen = True) then begin
    writeln(sum,'IMPEDANCE   = ',ZAvg:10:4);
    writeln(sum,'PHASE ANGLE = ',TAvg:10:4);
    writeln(sum,'DC BIAS     = ',DAvg:10:4);
    writeln(sum)
  end;

  a := chr(39);
  GetFileName(filenm,'Z',sequence);
  assign(f,status.path+PlotPath+filenm+'.pl');
  rewrite(f);
  writeln(f,'%      Electrical Data Plot Generator 2');
  writeln(f,'       '+status.path+ivpath+status.filebase+'.iv');
  writeln(f,'%      Timothy J. Dalton (c) 1991');
```

```
writeln(f);
writeln(f,'set page format portrait');

writeln(f,'set window x 1.0 to 7.5 y 1.0 to 3.333');
writeln(f,'title left '+a+'DC Bias (Volts)'+a);
writeln(f,'set order x dummy dummy dummy dummy dummy y');
if (status.lifer = false) then
  writeln(f,'set axis all on')
else
  writeln(f,'set axis all on right off');
writeln(f,'set axis left on top on');
writeln(f,'set limits x 0 ',endtime:6);
if (numpts > 1024) then begin
  I := round(numpts);
  j := I div 1024;
  writeln(f,'set sample ',(j+1):2);
end;
writeln(f,'insert '+status.path+ivpath+status.filebase+'.iv');
if (status.lifer = false) then
  writeln(f,'title bottom '+a+'Time (Sec)   ',
         'Avg[',0.40*runtime:6:1,',',0.60*runtime:6:1,'sec] = ',
          DAvg:10:4,a);
writeln(f,'draw axes');
writeln(f,'join');
if (Status.lifer = true) then begin
  writeln(f,'set order x y');
  writeln(f,'set limits x 0 ',endtime:4);
  writeln(f,'set axis all off right on');
  writeln(f,'set labels right on');
  writeln(f,'set sample 1');
  writeln(f,'title right '+a+'Interferometry Signal(v)'+a);
  writeln(f,'title bottom '+a+'Time (Sec)   ',
         'Avg[',0.40*runtime:6:1,',',0.60*runtime:6:1,'sec] = ',
          DAvg:10:4,a);
  if (numlifpts > 1024) then begin
    I := round(numlifpts);
    j := I div 1024;
    writeln(f,'set sample ',(j+1):2);
  end;

  writeln(f,'insert '+status.path+liferpath+status.filebase+'.lin');
  writeln(f,'set limits ymin default ymax default');
  writeln(f,'draw axes');
  writeln(f,'join dots');
end;

writeln(f,'set window x 1.0 to 7.5 y 4.333 to 6.666');
writeln(f,'title left '+a+'Phase Angle (Deg.)'+a);
writeln(f,'set order x dummy dummy dummy dummy y');
if (status.lifer = false) then
  writeln(f,'set axis all on')
else
  writeln(f,'set axis all on right off');
writeln(f,'set axis left on top on');
```

```
writeln(f,'set limits x 0 ',endtime:6);

writeln(f,'set limits y -90 -80 ');

if (numpts > 1024) then begin
  I := round(numpts);
  j := I div 1024;
  writeln(f,'set sample ',(j+1):2);
end;
writeln(f,'insert '+status.path+ivpath+status.filebase+'.iv');
if (status.lifer = false) then
  writeln(f,'title bottom '+a+'Time (Sec)   ',
         'Avg[',0.40*runtime:6:1,',',0.60*runtime:6:1,'sec] = ',
          TAvg:10:4,a);
writeln(f,'draw axes');
writeln(f,'join');
if (Status.lifer = true) then begin
  writeln(f,'set order x y');
  writeln(f,'set limits x 0 ',endtime:4);
  writeln(f,'set axis all off right on');
  writeln(f,'set labels right on');
  writeln(f,'set sample 1');
  writeln(f,'title right '+a+'Interferometry Signal(v)'+a);
  writeln(f,'title bottom '+a+'Time (Sec)   ',
         'Avg[',0.40*runtime:6:1,',',0.60*runtime:6:1,'sec] = ',
          TAvg:10:4,a);
  if (numlifpts > 1024) then begin
    I := round(numlifpts);
    j := I div 1024;
    writeln(f,'set sample ',(j+1):2);
  end;

  writeln(f,'insert '+status.path+liferpath+status.filebase+'.lin');
  writeln(f,'set limits ymin default ymax default');
  writeln(f,'draw axes');
  writeln(f,'join dots');
end;

writeln(f,'set window x 1.0 to 7.5 y 7.666 to 10.000');
writeln(f,'title left '+a+'Plasma Impedance (Ohms)'+a);
writeln(f,'title top '+a+status.filebase+a);
writeln(f,'set order x dummy dummy dummy y');
if (status.lifer = false) then
  writeln(f,'set axis all on')
else
  writeln(f,'set axis all on right off');
writeln(f,'set axis left on top on');
writeln(f,'set limits x 0 ',endtime:6);
writeln(f,'set limits y 0 50');

if (numpts > 1024) then begin
  I := round(numpts);
  j := I div 1024;
  writeln(f,'set sample ',(j+1):2);
```

```
    end;
    writeln(f,'insert '+status.path+ivpath+status.filebase+'.iv');
    if (status.lifer = false) then
      writeln(f,'title bottom '+a+'Time (Sec)   ',
              'Avg[',0.40*runtime:6:1,',',0.60*runtime:6:1,'sec] = ',
               ZAvg:10:4,a);
    writeln(f,'draw axes');
    writeln(f,'join');
    if (Status.lifer = true) then begin
      writeln(f,'set order x y');
      writeln(f,'set limits x 0 ',endtime:4);
      writeln(f,'set axis all off right on');
      writeln(f,'set labels right on');
      writeln(f,'set sample 1');
      writeln(f,'title right '+a+'Interferometry Signal(v)'+a);
      writeln(f,'title bottom '+a+'Time (Sec)   ',
              'Avg[',0.40*runtime:6:1,',',0.60*runtime:6:1,'sec] = ',
               ZAvg:10:4,a);
      if (numlifpts > 1024) then begin
        I := round(numlifpts);
        j := I div 1024;
        writeln(f,'set sample ',(j+1):2);
      end;

      writeln(f,'insert '+status.path+liferpath+status.filebase+'.lin');
      writeln(f,'set limits ymin default ymax default');
      writeln(f,'draw axes');
      writeln(f,'join dots');
    end;

    writeln(f,'stamp bottom date time');
    close(f);
end;


begin
end.
```

```
unit msoptvg;

{ MSOPT Variables, Global }

Interface

Const
 FNames : Array[1..5] of string = ('Temp1.TMP', 'Temp2.TMP',
 'Temp3.TMP', 'Temp4.TMP', 'Temp5.TMP');

  MaxPerSet = 20;
  MaxPeak  = 80;                  { Max. Number of Peaks to do }

TYPE
  ActinRec = Record
    GasName     : String[20];       { Name }
    Peak        : Real;             { Peak Location on Graph }
    Delta       : Real;             { Tolerance to Peak Location }
    PeakCtr     : Integer;          { Peak to Associate This With }
    WavePeak : Integer;  { Peak Wave Loc. if not assigned }
    Ratio       : Real;
  end;


VAR

  ACT           : Array[1..MaxPeak] of ActinRec;
  FS            : Real;  { Full Scale Data Range }
   N            : Integer;          { Current Point Number }
  PStart,
  PEnd,
  Pmax          : Array[1..MaxPeak] of Integer;
  PArea : Array[1..MaxPeak] of Real;
  NumberPeaks   : Integer;
  NumPeak       : Integer;
  NumberSpectra: Integer;
  NoRun, I, J,
  K, II   : Integer;
  Temp          : Word;
  Out           : Integer;
  FileName      : String;
  Dummy         : String;
  Tens          : Integer;
  Ch            : Char;
  Time          : Array[1..MaxPerSet] of real;
  TimeG         : Real;
  Code          : Integer;
  OutFile       : Text;
  Out2          : Text;
  Out3          : Text;
  SUM           : Text;
  SumOpen       : Boolean;
  AutoBase      : Boolean;
  Files   : Array[1..5] of text;
```

```
Resp            : Char;
PlotIt  : Char;
PlotSpectra  : Char;
RunString       : String[2];
AnalyzeSpectra : Boolean;
Actin  : Boolean;
PeakStart    : Real;                    { Peak Start Yet ?? }
PeakEnd          : Real;
SigmaPeak       : Real;                 { Still in a Peak ?? }
SlopeCriteria : Real;
XXPos, YYPos : Word;
n1,n2,lambda,theta : real;
endtime,RunTime : Real;
DataFilter : Boolean;
NumFilter:Integer;
CalcER          : Boolean;
NumLifPts       : Integer;


Implementation

begin
end.
```

```
unit lifanal;

{ Laser Interferometry Analysis }

INTERFACE
Uses Dos, Crt, Vars, Misc, MSOPTVG;

Type
  DatType        = Array[1..5] of real;

var
  A0, A1,
  R2, Syx,
  endtime,
  RunTime        : Real;

PROCEDURE CalcRSQ(x,y : DatType; Var A0, A1, R2, Syx : Real);
PROCEDURE GetTimes;

IMPLEMENTATION

PROCEDURE Pop(var x:dattype);
var I : integer;
begin
 for I := 2 to 5 do
   x[I-1] := x[I];
end;

PROCEDURE CalcRSQ(x,y : dattype; Var A0, A1, R2, Syx : Real);
var
 SumX, SumY, SUMXY, SUMX2, XMean, YMean, St, Sr,N : Real;
 I : Integer;

begin
 N := 5;
 SumX := 0.0;
 SumY := 0.0;
 SumX2 := 0.0;
 SumXY := 0.0;
 St := 0.0;
 Sr := 0.0;
 for I := 1 to round(N) do begin
   SumX  := Sumx  + X[i];
   SumY  := SumY  + Y[i];
   SumXY := SumXY + (X[i]*Y[i]);
   SumX2 := SumX2 + (X[i]*X[i]);
 end;
 XMean := SumX/N;
 YMean := SumY/N;
 A1 := (N*SumXY-SumX*SumY)/(N*SumX2-SumX*SumX);
 A0 := YMean - A1 * XMean;
 For I := 1 to round(N) do begin
  St := St + Sqr(Y[i] - YMean);
  Sr := Sr + Sqr(Y[i]-A1*X[i]-A0);
```

```
  End;
  R2 := (St-Sr)/St;
  SYX := Sqrt(Sr/(N-2));
end;

PROCEDURE GetTimes;

var
  f       : Text;
  f2      : text;
  T,V,
  Tavg : Real;
  num,
  i       : integer;
  dtime,
  dvolt,
  rsq,
  A1S   : DatType;
  dummy:real;
  r2old: Real;
  init    : real;

begin
  { Count the Number of Points }
  NumberExtrema := 0;
  assign(f,Status.path+LiferPath+status.filebase+'.LIN');
  reset(f);

  {Determine Run Time and EndTime for Graph Axis Limit }
  num := 0;
  while not(eof(f)) do begin
    readln(f, t,v);
    num := num + 1;
  end;
  RunTime := t;
  endtime := 10.0*int(t/10.0 +1.0);  { Round it up to next mult. of 10 }
  close(f);

  { Calc A linear least squares fit to 5 points in a row }
  reset(f);
  assign(f2, status.path+'/tp/temp.dat');
  rewrite(f2);
  for i := 1 to 5 do
    readln(f, dtime[i], dvolt[i]);
  CalcRSQ(dtime, dvolt, A0, A1, R2, Syx);
  tavg := 0; for j := 1 to 5 do tavg := tavg + dtime[j]; tavg := tavg/5.;
  writeln(f2,tavg:10:4,' ',r2:12:6,'  ',a1:12:6,'  ',a0:12:6,'  ',i,'  ',num);
  repeat
    r2old := r2;
    pop(dtime);
    pop(dvolt);
    readln(f, dtime[5], dvolt[5]);
    I := I + 1;
    CalcRSQ(dtime, dvolt, A0, A1, R2, Syx);
```

```
    tavg := 0; for j := 1 to 5 do tavg := tavg + dtime[j]; tavg := tavg/5.;
    writeln(f2,tavg:10:4,' ',r2:12:6,' ',a1:12:6,' ',a0:12:6,' ',i,' ',num);
  until (i = (Num) );
  close(f);
  close(f2);

  IF (CalcER = TRUE) then begin
    { Find Extrema }
    reset(f2);
    reset(f);
    {Skip 1st 2 pts as they are taken into the avg time}
    readln(f,dummy); readln(f,dummy);
    init := 0.0;
    for i := 1 to 5 do begin
      readln( f2, dtime[i], rsq[i], a1s[i] );
      readln( f, dummy,   dvolt[i] );
      init := init + dvolt[i];
    end;
    init := init / 5;
    while (not(eof(f2))) do begin
     pop(dtime);
     pop(rsq);
     pop(dvolt);
     pop(a1s);
     i := I + 1;
     readln(f2, dtime[5], rsq[5], a1s[5]);
     readln(f, dummy, dvolt[5] );
     { Are the signs of 1 & 2 the same,
       and the signs of 3 & 4 the same,
       and the signs of 1 and 3 different
       and RSquared < 0.75
       and deviation from baseline significant ? }
     if ( (  ( (A1S[1]*A1S[2] > 0.0) and
             (A1S[3]*A1S[4] > 0.0) and
             (A1S[1]*A1S[3] < 0.0) and
             (Rsq[3] < 0.75) and
             (abs(dvolt[3]-init) > 0.5)))) then begin
         NumberExtrema := NumberExtrema + 1;
         ExtremaTimes[NumberExtrema] := dtime[3];
         if (A1S[1] > 0.0) then
           ExtremaType[NumberExtrema] := 'Max'
         else
           ExtremaType[NumberExtrema] := 'Min';
      end; {if slope}
    end; {do while}
    close(f2);
    close(f);
  end {If CalcER}
  else
   NumberExtrema := 0;
end;

begin end.
```

Appendix 3

Full Wafer Interferometry Code

This appendix contains listings of the source code used for data acquisition and analysis of CCD images as part of the FWI Technique. All of this code for gathering data and choosing points was written in Microsoft C 7.0 and run on an 80386 PC. The analysis software was written in Unix Fortran and run on a Mips M120/5. The following list outlines the programs that are listed in this section, and their function.

Dsave2.c:	Gather CCD Images	603
Danal.c:	Display Image, Choose Points, Create Input Files for Analysis	614
ccdanal2.c700.f:	Determine Etch Rates	662

The listings for each of these units follows.

```
// dsave2.c
//
// t.j.dalton MIT 66-225 Cambridge MA 02139 (617) 258-8840
//
// del dsave2.obj
// cl /c /G2 /FPi87 /AH dsave2.c
// link dsave2.obj+highcam.obj+d4000.obj,dsave2.exe,NUL,,NUL
//
// take ccd images and save to ram disk.
// synchronize to magnetic field if desired.
//
// based on hsample.c from Electrim Corp.
// Sample C code to demonstrate use of HIGHCAM routine        11/11/91
// to read EDC-1000HR

// Compiled with Microsoft C 7.00

// Revised 01/10/92 to add user defined base address of camera

// Revised 03/01/92 to call routines in D4000.OBJ to display image on TJD
//              TSENG LABS ET4000 chipset SUPER VGA (with 1 meg. ram)

// Revised 03/26/92 to allow image save and exit. TJD

// Revised 04/06/92 to simplify buffer allocation

// Revised 05/23/92 to synchronize to magnetic field.

// Revised 11/02/93 to allow for data collection to trigger on external synch

// Revised 11/16/93 to allow for a slowing down of data acquisition rate
//          by setting the acq_speed;

// Revised 01/06/94 to set anti-blooming.

// Note: the program requires a TSENG LABS ET4000 chipset SUPER VGA to run,
//      but the sample code for reading the camera is general.
//      Use of HIGHCAM routine requires 80286 or better processor.


#include <stdlib.h>
#include <malloc.h>
#include <conio.h>
#include <dos.h>
#include <stdio.h>
#include <conio.h>


#define BASE     (0x360)               /* Base address of camera */
                                 /* must match camera address */
#define V       (244)              /* Lines per field */
#define H       (800)              /* Bytes returned per line */

#define TRUE    1
```

```
#define FALSE   0

#define DEFAULT_EXPOSURE 200L              /* 200 millisecond exposure */

typedef unsigned char pixel;               /* one byte per pixel */


// Linkable Rourines Function Prototypes:

unsigned short int _cdecl
highcam (unsigned int, int, int, int, int, unsigned long int, pixel _far * _near *);

int _cdecl _far
init800 (void);                            /* init vga to 800 x 600 */

void _cdecl _far
display_image (pixel _far * _near *);      /* display image buffer */


// HIGHCAM argument definitions:

// 0   normal return
// !0  a key was pressed during exposure time; buffer contents unmodified.



unsigned int base = BASE;                  /* base address of camera  */

// Flags:

int refresh_flag = !0;                     /* 0 disables refresh      */
int ab_flag = !0;                          /* 0 disables anti-blooming */

int interlace_flag = !0;                   /* !0 for interlace mode   */

int data_collect_synch_flag = 0;
int data_collect_synch = 0;                /* Synchronize Start of Data
                                              Collection Externally ?  */

int field_flag = 0;                        /* if interlace mode,      */
                                           /* 0 for first frame,      */
                                           /* !0 for second frame     */

unsigned long                   /* exposure time value in milliseconds */
exposure_time = DEFAULT_EXPOSURE;  /* for EDC-1000HR exposure control     */

pixel           _far * _near buffer[244];       /* image buffer */
pixel           _far * _near buffer2[244];      /* image buffer */
FILE            *fout;
int             i, j, keyhit, filenum;
int             inter, intnum;
int             mag_sync, ready;
int             speed_control = 0;        /* Control Acq Rate ? */
float           acq_speed = 0.0;  /* Speed to Acquire At */
```

```
unsigned        mag_field;
int             num_rise, prev_rise, prev_field;
unsigned        control_reg, port_1, port_3;
union REGS      inregs1, outregs1, inregs2, outregs2;
long            itime, ncalls;
float           time0, tttime;
double          tt;                     /* Time of Last Image */


//****************************************************************************

// Note:

// The image buffer pointer is an array of 244 pointers to buffers for each
// individual line in a field returned from the EDC-1000HR. For each line
// HIGHCAM returns 800 bytes. The first 44 bytes are non-image data, and
// may be disregarded.


//****************************************************************************


//****************************************************************************


// INIT800 return code definitions:

// 0      normal return, initialization for 800 x 600 mode completed
// 1      TSENG ET4000 VGA chip set not detected
// other  TSENG LABS VGA chip set not detected


// Note:  no check is made for the amount of memory on the VGA card.
//        one meg. of ram is required for proper operation.


//****************************************************************************



//****************************************************************************


// DISPLAY_IMAGE argument definition:

// The display buffers are as defined for HIGHCAM, but each line is displayed
// to the screen twice in non-interlace mode to give 488 lines total.


// Define storage for display buffer line pointers:

pixel _far * _near display_buffer[488];        /* display buffer */
```

```
//****************************************************************************

float gettime(void)
{
float temp;
struct dostime_t time0;

   _dos_gettime( &time0);
   temp = (float)time0.hour*3600 + (float)time0.minute*60 +
        (float)time0.second + (float)time0.hsecond*0.01;
   return(temp);
}


//****************************************************************************
void grclr(void)
{

  intnum=16;
  inregs1.x.ax=3;
  inter = int86(intnum, &inregs1, &outregs1);

}


//****************************************************************************
void grinit(void)
{
int TSENG_4000;

   if ((TSENG_4000 = init800 ()) != 0) {         /* init 800 x 600 mode */
          if (TSENG_4000 == 1)
             cprintf ("\r\nTSENG 4000 VGA chip set not detected!");
          else
             cprintf ("\r\nTSENG LABS VGA chip set not detected!");
          exit (-1);
   }
}


//****************************************************************************
void Check_Sync(void)
{
   data_collect_synch = ( _inp(port_1) & 2);
}

//****************************************************************************

void check_field(void)

/*
        Attempt to synchronize to magnetic field using
        microswitch ss21pe digital magnetic field sensor.
```

```
        Reverse logic.

        ss21pe output:    +5V    : No Field
                          +0V    : Field (15-25 Gauss Switch Range)

        Sensor draws power from dt2817 pin 20,
        ground is pin 22 and signal is pin 4 (port 1, bit 0)

        TJD 5/23/92
*/

{

  /* Look for a transition from low (field present) to
     high (field absent) and mark it! */

  prev_rise = num_rise;
  mag_field = ( _inp(port_1) & 1);
  if ( (mag_field == 1) && (prev_field == 0)) num_rise++;
  prev_field = mag_field;

} /* check_field */

//****************************************************************************


void saveimage(void)
{
int i,j;
size_t written;
char fname[80];

  filenum++;

  sprintf(fname, "d:\\image%d.dat",filenum);

  fout = fopen(fname,"wb");
  if (fout == NULL) {
        grclr();
        printf("ERROR OPENING OUTPUT FILE!\n");
        exit(1);
  }
  tt = (double) (tttime - time0);
  /* printf("%s %f \n",fname,tt); */
  for (i=0; i<V; i++) {
    if (mag_sync == TRUE) check_field();
    written = fwrite(&(buffer[i][44]), (size_t) sizeof(pixel), (size_t) H-44,
          fout);
    if (written < (size_t)H-44) {
//      printf("WRITTEN = %d \n",(int) written);
      grclr();
      printf("DSAVE Execution terminated.\n");
      exit(1); }
    if (interlace_flag != 0) {
```

```
      if (mag_sync == TRUE) check_field();
      written = fwrite(&(buffer2[i][44]), (size_t) sizeof(pixel),
             (size_t) H-44, fout);
      if (written < (size_t)H-44) {
//        printf("WRITTEN = %d \n",(int) written);
            grclr();
        printf("DSAVE Execution terminated.\n");
        exit(1); }
    } /* IF */
  }
  written = fwrite(&tt, (size_t) sizeof(double), (size_t) 1, fout);

 fclose(fout);

} /* SAVE IMAGE */


void readcam(void)
{

// Read the camera:

   if (mag_sync == TRUE) check_field();
   highcam (base, refresh_flag, ab_flag, interlace_flag, 0,
          exposure_time, buffer);
   if (mag_sync == TRUE) check_field();
   if (interlace_flag != 0)
     highcam (base, refresh_flag, ab_flag, interlace_flag, !0,
            exposure_time, buffer2);
   if (mag_sync == TRUE) check_field();
}


void _cdecl
main (void)
{

   int TSENG_4000;
   int i,j,saveit,end_prog;
   char resp;

control_reg      = 0x228;
port_1   = 0x22a;
port_3   = 0x22c;

printf("Interlaced Mode ? (Y/N) : ");
scanf("%c",&resp);
if ((resp=='y') || (resp=='Y'))
  interlace_flag = !0;
else
  interlace_flag = 0;

// Allocate the image buffers:
```

```
 for (i=0; i<V; i++) {

    if ((buffer[i] =
       (pixel _far *) calloc ((size_t) H, sizeof (pixel)))
       == (pixel _far *) NULL) {

       cprintf ("\n\n\rCannot allocate memory for buffer!\n\r");
       exit ((char) -1);
    }
 } /* FOR i */

 for (i=0; i<V; i++) {

    if ((buffer2[i] =
       (pixel _far *) calloc ((size_t) H, sizeof (pixel)))
       == (pixel _far *) NULL) {

       cprintf ("\n\n\rCannot allocate memory for buffer2!\n\r");
       exit ((char) -1);
    }
 } /* FOR i */


printf("Enter the Exposure Time in mSec: ");
scanf("%d",&exposure_time);
printf("\nExposure Time = %d\n",exposure_time);

printf("\nSynchronize to Magnetic Field Rotation ? (Y/N) : ");

do {
 keyhit = kbhit();
 if ( (keyhit=kbhit() ) != 0) keyhit = getch();
}
while (!(keyhit == 'y' || keyhit == 'Y' || keyhit == 'n' || keyhit == 'N'));

resp = keyhit;

if ((resp=='y') || (resp=='Y'))
  {
       printf("YES\n\n");
    mag_sync = TRUE;
    _outp( control_reg, 0);          /* Set all ports to input */

       /* The magnetic field sensor will go from low to
          high twice during a period. We want to begin a new
          scan when on the second rise.
              +-+      +-+      +-
              | |      | |      |
              | |      | |      |
          ----+ +------+ +------+
                ^                ^
               t               t+T

          We have to keep track (even while saving)
          of the status of the magnetic field so that
```

```
            we trigger on the 2nd (or 4th) field.

          */
    }
  else
    {
      printf("NO\n\n");
      mag_sync = FALSE;
          printf("\n\n0 - Maximum Throughput\n");
          printf("1 - Controled Rate\n\n");
          printf("Enter Data Acquisition Rate:  ");
          scanf("%d",&speed_control);
          if ((speed_control != 0) && (speed_control != 1))
            speed_control = 0;
          if (speed_control == 1)  {
            printf("\n\nEnter Time in Sec. between images:  ");
            scanf("%f",&acq_speed);
            printf("\nData Will be taken ever %f seconds.\n",acq_speed);
      }
    }


  printf("\nActivate Anti-Blooming on CCD ? (Y/N) : ");

  do {
   keyhit = kbhit();
   if ( (keyhit=kbhit() ) != 0) keyhit = getch();
  }
  while (!(keyhit == 'y' || keyhit == 'Y' || keyhit == 'n' || keyhit == 'N'));

  resp = keyhit;

  if ((resp=='y') || (resp=='Y'))
    {
      ab_flag = !0;                                     /* 0 disables anti-blooming */
          printf("Anti-Blooming Activated\n");
    }
  else
    {
          ab_flag = 0;
          printf("Anti-Blooming NOT Activted\n");
    }


// Build display buffers from image buffers:


   for (i=0, j=0; i<V; i++, j+=2) {
      display_buffer[j]  = buffer[i];
      if (interlace_flag != 0)
            display_buffer[j+1] = buffer2[i];
          else
            display_buffer[j+1] = buffer[i];
   }
```

```
filenum = 0;
saveit = 0;

printf("\n\nBEGINNING PROGRAM\n\n");
printf("Press Choice: \n\n");
printf(" S  - Save\n E  - Start on External Synch\n Q  - Quit\nESC - Stop Save\n");
end_prog = FALSE;

do {


  /* Should we save it while looping ? and if so,
        Should we worry about the magnetic field period ? */

  if (saveit == 1) {
        if (mag_sync == TRUE) {

          /* We have to Wait for a change in num_rise and
            a value of num_rise of 1, 3, 5, 7, etc. */

            ready = FALSE;
            do {
                  check_field();
                  if ( ( (num_rise%2) == 1 ) && (num_rise != prev_rise) )
                    ready = TRUE;

                    if ( (keyhit=kbhit() ) != 0) {
                    keyhit = getch();
                              if ( (keyhit == 'q') || (keyhit == 'Q') ) {
                                grclr();
                                printf("DSAVE Execution terminated.\n");
                                end_prog = TRUE;
                                exit(1);
                              }
                    }

            } while (!ready);

            /* Now, We can read and Save it! */
        tttime = gettime();
            readcam();
            saveimage();
            printf("%d ",filenum);
          }
        else {

          if (speed_control == 0) {
          tttime = gettime();
        readcam();
                  saveimage();
                  printf("%d ",filenum);
          } else {
                  if (filenum != 0)
                    do { } while ( (gettime() - time0) < (tt + acq_speed) );
```

```
          tttime = gettime();
        readcam();
                saveimage();
                printf("%d ",filenum);
          }
        }
    } /* if SAVEIT */

      display_image (display_buffer);

        /* If We're saving in the external synch mode, and then the
          Synch signal is lost, that we are done. */
        if ((saveit == 1) && (data_collect_synch_flag == 1)) {
                Check_Sync();
                if ( data_collect_synch == 0)
              end_prog = TRUE;

        }

    if ( (keyhit=kbhit() ) != 0) {
      keyhit = getch();
      switch (keyhit) {

          /* Quit */
          case 'q':
          case 'Q':         end_prog = TRUE;
                            break;

          /* Start Data Collection on External Synch */
          case 'e':
          case 'E':         /* First, Wait for Flag to go High */ {
                            grinit(); grclr();
                            printf("\n\nWaiting For External Synch\n\n");
//                          printf("Press Return to Begin Wait.\n");
//                          do {} while ( (keyhit=kbhit() ) == 0);

                            data_collect_synch_flag = 1;
                            do
                              Check_Sync();
                            while (data_collect_synch == 0);
                            printf("External Synch Arrived\n\n");
                            }
                            if (saveit == 0) {
//                            printf("SAVING!\n");
                              grinit();
                              saveit = 1;
                              if (mag_sync == TRUE)
                                 prev_field = ( _inp(port_1) & 1);
                              num_rise = 0; prev_rise = 0;
                              time0 = gettime();
                            }
                            break;

          /* Save Data */
```

```
        case 's':
            case 'S':       if (saveit == 0) {
//                            printf("SAVING!\n");
                              grinit();

                              saveit = 1;
                              if (mag_sync == TRUE)
                                 prev_field = ( _inp(port_1) & 1);
                              num_rise = 0; prev_rise = 0;
                              time0 = gettime();
                            }
                            break;

            /* Escape */
        case  27:   if (saveit == 1) {
                              saveit = 0;
//                            printf("STOPPING SAVE With %d Images\n",filenum);
                            }
                            break;
      } /* SWITCH */
    } /* IF */


  } while ( !(end_prog));

  grclr();
  printf("DSAVE Execution terminated.\n");

} /* MAIN */
```

```
/ Danal.c
/* Disk Analysis of CCD Images. */

// Compiled with Microsoft C 7.00
// del danal.obj
// cl /c /G2 /FPi87 /AH danal.c
// link danal.obj+highcam.obj+d4000.obj,danal.exe,NUL,graphics.lib+lmouse.lib,NUL

// Revised 01/10/92 to add user defined base address of camera

// Revised 03/01/92 to call routines in D4000.OBJ to display image on
//              TSENG LABS ET4000 chipset SUPER VGA (with 1 meg. ram)

// Revised 03/26/92 to allow image save and exit. TJD

// Revised 04/06/92 to simplify buffer allocation

// Note:  the program requires a TSENG LABS ET4000 chipset SUPER VGA to run,
//      but the sample code for reading the camera is general.

//      Use of HIGHCAM routine requires 80286 or better processor.


#include <stdlib.h>
#include <malloc.h>
#include <conio.h>
#include <dos.h>
#include <stdio.h>
#include <conio.h>
#include <math.h>
#include <graph.h>
#include <mouse.h>


#define BASE      (0x360)              /* Base address of camera */
                              /* must match camera address */

#define V       (244)              /* Lines per field */
#define H       (800)              /* Bytes returned per line */

#define DEFAULT_EXPOSURE 200L            /* 200 millisecond exposure */

#define RGB(r,g,b) (0x3F3F3FL & ((long)(b) << 16 | (g) << 8 | (r)))
#define MAXCOL                 256                    /* Maximum Colors */
#define TRUE            1
#define FALSE           0

#define         NMAX  (4)
#define         PTMAX         (256)
#define         FMAX  (8196)  /* PTMAX * 2^3 *2 +1   + 3 safety */
#define         NNMAX         (12)


typedef unsigned char pixel;            /* one byte per pixel */
```

```
// Linkable Rourines Function Prototypes:


unsigned short int _cdecl
highcam (unsigned int, int, int, int, int, unsigned long int, pixel _far * _near *);


int _cdecl _far
init800 (void);                          /* init vga to 800 x 600 */


void _cdecl _far
display_image (pixel _far * _near *);       /* display image buffer */

//******************************************************************************

// HIGHCAM return value definitions:

// 0  normal return
// !0  a key was pressed during exposure time; buffer contents unmodified.


//******************************************************************************



unsigned int base = BASE;                /* base address of camera  */

// Flags:

int refresh_flag = !0;                  /* 0 disables refresh      */
int ab_flag = 0;                       /* 0 disables anti-blooming */
int interlace_flag = !0;                  /* !0 for interlace mode   */
int field_flag = 0;                      /* if interlace mode,      */
                                   /* 0 for first frame,      */
                                   /* !0 for second frame     */



unsigned long                /* exposure time value in milliseconds */
exposure_time = DEFAULT_EXPOSURE;  /* for EDC-1000HR exposure control    */


pixel _far * _near buffer[244];          /* image buffer */
pixel _far * _near buffer2[244];          /* image buffer */
FILE *fout,*fsum;
int i, j, keyhit, filenum;
int inter,intnum;
union REGS inregs1, outregs1, inregs2, outregs2;
long itime,ncalls;
int num_anal_pts;
#define  MAXANALPTS  800
int anal_type[MAXANALPTS], anal_ptsx[MAXANALPTS], anal_ptsy[MAXANALPTS],
```

```
   anal_sizex[MAXANALPTS], anal_sizey[MAXANALPTS];
float analavg[MAXANALPTS];
int centx, centy;
int initnum;
float mapx, mapy;
char fpath[80], fblank[80];
double tt;
int rotate, contr;

struct palette {
  pixel   Red[MAXCOL];
  pixel   Grn[MAXCOL];
  pixel   Blu[MAXCOL];
} MasPal;

long imsize;
char __huge *buf;

static char *grid_type[] = {"Series of Rectangular Regions",
                            "Rectangular Area Array       ",
                               "Round Area Array          ",
                               "Line Array               "};
#define         RECT_REGIONS            0
#define         RECT_AREA_ARRAY         1
#define         ROUND_AREA_ARRAY        2
#define         LINE_ARRAY          3

int grid_num = RECT_REGIONS;

long pal[256];
short red, blue, green;
short inc, mode, cells, x, y, xinc, yinc;
int done, xlim, ylim, key, dx, dy, ox, oy, xxmin, yymin, ii, xx, yy;
int bn, bx, by, prev_bn, prev_bx, prev_by;
int lx, ly, rx, ry, tx, ty, box, boy, centered;
int x0, yy0, x2, y2, x3, y3, x4, y4;
float theta2;
char buf2[40];
struct _videoconfig vc;
struct _fontinfo fi;

int     cnt, start, nn, actpts, pow2;
int     numpts;                 /* Actual Number of Analysis Points */
int     dum[1];                 /* Dummy Variable */
int     fmaxi[NMAX];            /* Maximum Freq Location */
float   xpix, ypix, theta; /* Wafer Scaling & Orientation */
float   fpow2;                  /* Floating Power of 2 for FFT */
float   sum, sdelta;            /* Sampling Time */
float   r[NMAX], th[NMAX];      /* Radius and Angle */
float   t[PTMAX];               /* Time */
float   yv[PTMAX][NMAX];        /* y values */
float   ym[PTMAX];              /* Array of Y Maximums */
float   ycall[PTMAX];           /* Temp Array for Calls */
float   a1;                     /* Intercept of Fit Line */
```

```
float    b1;                        /* Slope of Fit Line */
float    siga, sigb;                /* Std. Dev. of Fit */
float    chi2,q;                    /* Chi-Squared & Q for Fit */
float    freq[FMAX];                /* Frequencies */
float    fft[FMAX];                 /* FFT ARRAY */
float    fmag[FMAX][NMAX];          /* Magnitude of FFT */
float    fmax[NMAX];                /* Max. Frequency */
float    ymax, ymin, xmax, xmin;    /* For Graphs */
float    pi;                        /* PI */


//******************************************************************************

// Note:

// The image buffer pointer is an array of 244 pointers to buffers for each
// individual line in a field returned from the EDC-1000HR.  For each line
// HIGHCAM returns 800 bytes.  The first 44 bytes are non-image data, and
// may be disregarded.

//******************************************************************************


// INIT800 return code definitions:

// 0      normal return, initialization for 800 x 600 mode completed
// 1      TSENG ET4000 VGA chip set not detected
// other  TSENG LABS VGA chip set not detected


// Note:  no check is made for the amount of memory on the VGA card.
//        one meg. of ram is required for proper operation.

//******************************************************************************



//******************************************************************************


// DISPLAY_IMAGE argument definition:

// The display buffers are as defined for HIGHCAM, but each line is displayed
// to the screen twice in non-interlace mode to give 488 lines total.


// Define storage for display buffer line pointers:


pixel _far * _near display_buffer[488];        /* display buffer */
```

```
//***************************************************************************
void restore_screen(void)
{
 intnum=16;
 inregs1.x.ax=3;
 inter = int86(intnum, &inregs1, &outregs1);
}

void plot_anal_pts(void)
{
int i,j,k;

// printf("     plot_anal_pts: num_anal_pts = %d\n",num_anal_pts);
// for (i=0; i<num_anal_pts; i++) {
//   printf("     %d %d %d %d\n",anal_ptsx[i],anal_ptsy[i],anal_sizex[i],
//   anal_sizey[i]);
// }

  for (k=0; k<num_anal_pts; k++) {
    for (i=anal_ptsy[k]-anal_sizey[k]; i<=anal_ptsy[k]+anal_sizey[k]; i++) {
      for (j=anal_ptsx[k]-anal_sizex[k]; j<=anal_ptsx[k]+anal_sizex[k]; j++) {
//        printf("        i = %d   j = %d \n",i,j);
        buffer[i][j] = 255;
      } /* FOR J */
    } /* FOR I */
  } /* FOR K*/
}

void calc_radius(void)
{
int k, refl;
int dx, dy;
float ddx, ddy, d;
float ang;

  if (interlace_flag == 0)
    refl = H;
  else
    refl = 2*H;

   fprintf(fsum,"   ");
   for (k=0; k<num_anal_pts; k++) {
    dx  = centx - anal_ptsx[k];
    ddx = mapx * (float) dx / 1e4;     /* convert to cm */
    dy  = - (centy - anal_ptsy[k]);
    ddy = mapy * (float) dy / 1e4;
    d   = (float) sqrt( (double) (ddx*ddx+ddy*ddy));
    fprintf(fsum, "%6.3f ",d);
   } /* FOR */
   fprintf(fsum, "\n");

   fprintf(fsum,"   ");
   for (k=0; k<num_anal_pts; k++) {
    dx  = centx - anal_ptsx[k];
```

```
   ddx = mapx * (float) dx / 1e4;     /* convert to cm */
   dy  = - (centy - anal_ptsy[k]);
   ddy = mapy * (float) dy / 1e4;
   ang = (float) atan2( (double) ddy, (double) ddx);
   fprintf(fsum, "%6.1f ",ang*180./3.141592654);
  } /* FOR */
 fprintf(fsum, "\n");

}


void anal_wafer(void)
{
int i,j,k,num;
int ii,jj;
float avg;

// printf("      anal_wafer: num_anal_pts = %d\n",num_anal_pts);


 for (k=0; k<num_anal_pts; k++) {
   avg = 0.0;
   num = 0;
   for (i=anal_ptsy[k]-anal_sizey[k]; i<=anal_ptsy[k]+anal_sizey[k]; i++) {
     for (j=anal_ptsx[k]-anal_sizex[k]; j<=anal_ptsx[k]+anal_sizex[k]; j++) {
       num++;
           if (anal_type[k] == 0) {
             ii = i; jj = j; }
           else {
//
//           rotate the points so that we fit them into the
//           space better.
//
/*
             jj = j - anal_ptsx[k];
             ii = i - anal_ptsy[k];
             jj = (int)( jj * cos(theta2) - ii * sin(theta2));
             ii = (int)( jj * sin(theta2) + ii * cos(theta2));
             jj = jj + anal_ptsx[k];
             ii = ii + anal_ptsy[k];
*/
         ii = i;
         jj = j;

           }


       avg = avg + (float) buffer[ii][jj];
     } /* FOR J */
   } /* FOR I */

   analavg[k] = avg / (float) num;
   fprintf(fsum,"%6.1f ",analavg[k]);
```

```
  } /* FOR K*/
  fprintf(fsum,"\n");
}

void read_anal_points(endit)
int endit;
{
int i;

  if ( (fout = fopen("analpts.dat","r")) == NULL) {
          if (endit == TRUE) {
             restore_screen();
        printf("ERROR on analpts.dat open\n");
        exit(1); }
          else {
             centx = 400;
             centy = 100;
             mapx  = 150.0;
             mapy  = 300.0;
             theta = 0.0;
             }

      } /* IF NULL */
  else {

     fscanf(fout, "%d", &num_anal_pts);

     for (i=0; i<num_anal_pts; i++) {
       fscanf(fout,"%d %d %d %d  %d",&(anal_ptsx[i]), &(anal_ptsy[i]),
       &(anal_sizex[i]), &(anal_sizey[i]), &(anal_type[i]) );
     } /* FOR */
     fscanf(fout, "%d %d",&centx, &centy);
     fscanf(fout, "%f %f %f",&mapx, &mapy, &theta2);

     fclose(fout);
  } /* IF NULL */

} /* SHOW ANAL POINTS */


void show_anal_points(void)
{
  read_anal_points(TRUE);
  plot_anal_pts();

} /* SHOW ANAL POINTS */


//*****************************************************************************
void contrast(void)
{
register short int i,j,max,min;
float range;
// FILE *fout;
```

```
//   fout = fopen("contrast.dat","w");
  max = -1;
  min = 256;
  for (i=(44+10); i<(H-54); i++)
   for (j=(0+4); j<(V-3); j++)
        {
        if (buffer[j][i] > max) max = buffer[j][i];
        if (buffer[j][i] < min) min = buffer[j][i];
        }
  range = (float) (max-min);
// fprintf(fout,"Max = %d   Min = %d   Range = %f\n",max,min,range);
  for (i=(44+10); i<(H-54); i++)
   for (j=(0+4); j<(V-3); j++)
        {
        buffer[j][i] = (pixel) ( 255.0/range*(float)(buffer[j][i]-min) );
        }
// max = -1;
// min = 256;
// for (i=(44+10); i<(H-54); i++)
// for (j=(0+4); j<(V-3); j++)
// {
// if (buffer[j][i] > max) max = buffer[j][i];
// if (buffer[j][i] < min) min = buffer[j][i];
// }
// fprintf(fout,"Max = %d   Min = %d   Range = %f\n",max,min,range);
// fclose(fout);

} /* CONTRAST() */



//****************************************************************************
void tim_display_image(void)
{
register short int i,j;

 if (contr == TRUE) contrast();

 for (i=44; i<H; i++)
   for (j=0; j<V; j++)
        {
        _setcolor( buffer[j][i] );
        _setpixel( i, 2*j);
        _setpixel( i, 2*j+1);
        }
}


//****************************************************************************
void plotinit2(initnum)
int initnum;
{
int i,j;
long written;
```

```
char fname[80];

   sprintf(fname, fpath, initnum);

   fout = fopen(fname,"rb");
   if (fout == NULL)  return;

   for (i=0; i<V; i++) {
     written = fread(&(buffer[i][44]), (size_t) sizeof(pixel),
            (size_t) H-44, fout);
     if (written < (size_t)H-44) return;

     if (interlace_flag != 0) {
       written = fwrite(&(buffer2[i][44]), (size_t) sizeof(pixel),
              (size_t) H-44, fout);
       if (written < (size_t)H-44) return;
     }
   }

   written = fread(&tt, (size_t) sizeof(double), (size_t) 1, fout);
   if (written == 0)
         tt = (float) filenum;

   fclose(fout);
//   show_anal_points();
   tim_display_image();

} /* plot init */

void read_colors(void)
{

  int FixIt,i;
  FILE *fp, *fin;
  char PaletteName[80];

  fin = fopen("colormap.dat","r");
  if (fin == NULL) {
    _setvideomode( _DEFAULTMODE);
    printf("UNABLE TO LOCATE COLORMAP NAME!\n"); exit(1); }

  fscanf(fin,"%s",PaletteName);
  fclose(fin);

      if(( fp = fopen(PaletteName,"rb")) == NULL){
            _setvideomode( _DEFAULTMODE);
            printf("Error Opening Palette %s", PaletteName);
                 exit(1);
      }

      FixIt = fread(MasPal.Red, sizeof(pixel), 256, fp);
      FixIt = fread(MasPal.Grn, sizeof(pixel), 256, fp);
      FixIt = fread(MasPal.Blu, sizeof(pixel), 256, fp);
```

```
    fclose(fp);
        MasPal.Blu[0] = 0; MasPal.Grn[0] = 0; MasPal.Red[0] = 0;
        MasPal.Blu[255] = 255; MasPal.Grn[255] = 255; MasPal.Red[255] = 255;
}

//***************************************************************************

void two_color_scale(void)

{
register int i;
  read_colors();
  for (i=0; i<MAXCOL; i++)
  {
    pal[i]  = RGB(MasPal.Red[i]/4,MasPal.Grn[i]/4,MasPal.Blu[i]/4);
  }
  _remapallpalette(pal);
}



//***************************************************************************
void color_scale(void)

{
register int i,j;

/*
  read_colors();
  for (j=0,i=0; i<MAXCOL; i+=4,j++)
  {
    pal[j]  = RGB(MasPal.Red[i]/4,MasPal.Grn[i]/4,MasPal.Blu[i]/4);
    pal[64+j] = RGB(MasPal.Red[i+1]/4,MasPal.Grn[1+i]/4,MasPal.Blu[1+i]/4);
    pal[128+j] = RGB(MasPal.Red[i+2]/4,MasPal.Grn[2+i]/4,MasPal.Blu[2+i]/4);
    pal[192+j] = RGB(MasPal.Red[i+3]/4,MasPal.Grn[3+i]/4,MasPal.Blu[3+i]/4);
  }
*/

  read_colors();
  for (j=0,i=0; i<MAXCOL; i+=8,j++)
  {
    pal[   j] = RGB(MasPal.Red[i  ]/4,MasPal.Grn[i  ]/4,MasPal.Blu[i  ]/4);
    pal[ 32+j] = RGB(MasPal.Red[i+1]/4,MasPal.Grn[1+i]/4,MasPal.Blu[1+i]/4);
    pal[ 64+j] = RGB(MasPal.Red[i+2]/4,MasPal.Grn[2+i]/4,MasPal.Blu[2+i]/4);
    pal[ 96+j] = RGB(MasPal.Red[i+3]/4,MasPal.Grn[3+i]/4,MasPal.Blu[3+i]/4);
    pal[128+j] = RGB(MasPal.Red[i+4]/4,MasPal.Grn[4+i]/4,MasPal.Blu[4+i]/4);
    pal[160+j] = RGB(MasPal.Red[i+5]/4,MasPal.Grn[5+i]/4,MasPal.Blu[5+i]/4);
    pal[192+j] = RGB(MasPal.Red[i+6]/4,MasPal.Grn[6+i]/4,MasPal.Blu[6+i]/4);
    pal[224+j] = RGB(MasPal.Red[i+7]/4,MasPal.Grn[7+i]/4,MasPal.Blu[7+i]/4);
  }

/*
  read_colors();
  for (j=0,i=0; i<MAXCOL; i+=16,j++)
```

```
  {
   pal[   j] = RGB(MasPal.Red[i  ]/4,MasPal.Grn[i  ]/4,MasPal.Blu[i  ]/4);
   pal[ 16+j] = RGB(MasPal.Red[i+1]/4,MasPal.Grn[1+i]/4,MasPal.Blu[1+i]/4);
   pal[ 32+j] = RGB(MasPal.Red[i+2]/4,MasPal.Grn[2+i]/4,MasPal.Blu[2+i]/4);
   pal[ 48+j] = RGB(MasPal.Red[i+3]/4,MasPal.Grn[3+i]/4,MasPal.Blu[3+i]/4);
   pal[ 64+j] = RGB(MasPal.Red[i+4]/4,MasPal.Grn[4+i]/4,MasPal.Blu[4+i]/4);
   pal[ 80+j] = RGB(MasPal.Red[i+5]/4,MasPal.Grn[5+i]/4,MasPal.Blu[5+i]/4);
   pal[ 96+j] = RGB(MasPal.Red[i+6]/4,MasPal.Grn[6+i]/4,MasPal.Blu[6+i]/4);
   pal[112+j] = RGB(MasPal.Red[i+7]/4,MasPal.Grn[7+i]/4,MasPal.Blu[7+i]/4);
   pal[128+j] = RGB(MasPal.Red[i+8]/4,MasPal.Grn[8+i]/4,MasPal.Blu[8+i]/4);
   pal[144+j] = RGB(MasPal.Red[i+9]/4,MasPal.Grn[9+i]/4,MasPal.Blu[9+i]/4);
   pal[160+j] = RGB(MasPal.Red[i+10]/4,MasPal.Grn[10+i]/4,MasPal.Blu[10+i]/4);
   pal[176+j] = RGB(MasPal.Red[i+11]/4,MasPal.Grn[11+i]/4,MasPal.Blu[11+i]/4);
   pal[192+j] = RGB(MasPal.Red[i+12]/4,MasPal.Grn[12+i]/4,MasPal.Blu[12+i]/4);
   pal[208+j] = RGB(MasPal.Red[i+13]/4,MasPal.Grn[13+i]/4,MasPal.Blu[13+i]/4);
   pal[224+j] = RGB(MasPal.Red[i+14]/4,MasPal.Grn[14+i]/4,MasPal.Blu[14+i]/4);
   pal[240+j] = RGB(MasPal.Red[i+15]/4,MasPal.Grn[15+i]/4,MasPal.Blu[15+i]/4);
  }
*/


  _remapallpalette(pal);
}


//******************************************************************************
void gray_scale(void)

{
register int i;

FILE *fout;
fout = fopen("temp.pal","w");

  for (i=0; i<4*64; i+=4)
  {
    pal[i] = RGB(i/4,i/4,i/4);
    pal[i+1] = pal[i];
    pal[i+2] = pal[i];
    pal[i+3] = pal[i];
fprintf(fout,"%d  %d %d %d %d\n",i,pal[i],pal[i+1],pal[i+2],pal[i+3]);
  }
fclose(fout);
  _remapallpalette(pal);
}


//******************************************************************************
void plotinit(initnum)
int initnum;
{
int i,j;
size_t written;
char fname[80];
```

```
 filenum = initnum;

   sprintf(fname, fpath,filenum);

   fout = fopen(fname,"rb");
   if (fout == NULL)  return;

   for (i=0; i<V; i++) {
     written = fread(&(buffer[i][44]), (size_t) sizeof(pixel),
            (size_t) H-44, fout);
     if (written < (size_t)H-44) return;

     if (interlace_flag != 0) {
       written = fwrite(&(buffer2[i][44]), (size_t) sizeof(pixel),
              (size_t) H-44, fout);
       if (written < (size_t)H-44) return;
     }
   }

   written = fread(&tt, (size_t) sizeof(double), (size_t) 1, fout);
   if (written == 0)
         tt = (float) filenum;

   fclose(fout);
   if (contr == TRUE) contrast();
   show_anal_points();
   display_image (display_buffer);

} /* plot init */



void plotimage(void)
{
int i,j;
size_t written;
char fname[13];

 filenum = 0;

 do {
   filenum++;

   sprintf(fname, fpath, filenum);

   fout = fopen(fname,"rb");
   if (fout == NULL)  return;

   for (i=0; i<V; i++) {
     written = fread(&(buffer[i][44]), (size_t) sizeof(pixel),
            (size_t) H-44, fout);
     if (written < (size_t)H-44) return;

     if (interlace_flag != 0) {
```

```
        written = fwrite(&(buffer2[i][44]), (size_t) sizeof(pixel),
                (size_t) H-44, fout);
        if (written < (size_t)H-44) return;
      } /* IF */
    }
   written = fread(&tt, (size_t) sizeof(double), (size_t) 1, fout);
    if (written == 0)
          tt = (float) filenum;
//     printf("Read in File %d and time is %f\n",filenum, tt);

    fclose(fout);

    fprintf(fsum,"%6.2f ",tt);
    anal_wafer();
    plot_anal_pts();
    display_image (display_buffer);

  } while (1); /* DO */

} /* plot IMAGE */


/* ********************************************************************** */
/* Generic Rounding Routine */
int round(x)
  float x;
{
  if ( (x - (int)x) < 0.5 )
    return ((int)x);
  else
    return ((int)x+1);
}

/* ********************************************************************** */
void clear_bottom(void)
{
   _settextcolor(0);
   _settextposition(32,0);
   _outtext("                                                  ");
   _settextposition(32,60);
   _outtext("                              ");
   _settextposition(33,0);
   _outtext("                                                  ");
   _settextposition(33,60);
   _outtext("                              ");
   _settextposition(34,0);
   _outtext("                                                  ");
   _settextposition(35,0);
   _outtext("                                                  ");
   _settextposition(36,0);
   _outtext("                                                  ");
   _settextposition(37,0);
   _outtext("                                                  ");
}
```

```
/* ************************************************************************ */


/* ************************************************************************ */
void make_round_grid(void)
{
  float NumR, dTH, PI, Rmax, Rmin, R, TH, x, y, xx, yy, th2;
  int xxx,yyy, sizex, sizey;

  NumR = 20.0;
  Rmax = 5.00;   /* Cm */
  Rmin = 0.25;
  sizex = sizey = 1;

  dTH  = 10.0;

  PI  = 3.141592654;

  num_anal_pts = 0;
  for (R=Rmin; R<Rmax; R+=(Rmax-Rmin)/NumR) {
    for(TH=0.0; TH<360; TH += dTH) {
     th2 = TH * PI/180.0;
         x  = R*cos(th2+PI);
         y  = R*sin(th2+PI);
     xx = centx + x*1e4/mapx;
         xxx = round(xx) - sizex;
     yy = centy + y*1e4/mapy;
         yyy = round(yy) - sizey;
         if ( (xxx >= (44+10)) && (xxx <= (H-10)) &&
             (yyy >=  (0+2)) && (yyy <= (V-2)) ) {
            anal_ptsx[num_anal_pts]  = xxx;
            anal_ptsy[num_anal_pts]  = yyy;
            anal_sizex[num_anal_pts] = sizex;
            anal_sizey[num_anal_pts] = sizey;
            anal_type[num_anal_pts] = 0;
            num_anal_pts++;
         } /* IF */
    } /* FOR TH */
  } /* FOR R */

  grid_num = ROUND_AREA_ARRAY;

}


/* ************************************************************************ */
void make_line_grid(void)
{

char buf[80];
int done2, i, j;
int corner_num, corner_lim;
float xvals[2], yvals[2];
float xstart, xend, ystart, yend, NumX, NumY;
```

```
done2 = FALSE;

clear_bottom();
_settextcolor(255);
_settextposition(33,0);
_outtext("Line Grid Mode              \n");
_settextposition(34,0);
_outtext("(E)xit, (D)elete Point, (Q)uit.   RightMouse to Choose");
_settextposition(35,0);
_outtext("Left:");
corner_num = 0;
corner_lim = 2;
prev_bn = RIGHT;
_setwritemode(_GXOR);
xvals[1] = xvals[2] = 0.0;
yvals[1] = yvals[2] = 0.0;

do {
       switch (corner_num) {
         case 0:
                _settextposition(35,0);
                _outtext("Left:             ");
                break;
         case 1:
                _settextposition(35,40);
                _outtext("Right:            ");
                break;
         case 2:
                _settextposition(37,0);
                _outtext("Ready to (E)xit             ");
                break;
       } /* Switch corner_num */


   key = kbhit();
       GetMickeys(&xx, &yy);

   if ( (key != 0) || (xx != 0) || (yy != 0) ) {
         if ( (xx == 0) && (yy == 0) )
       key = getch();
         else
           key = 1;
         switch(key)
         {
           case 1:
           case 0:
           case 224:
             if ( (xx == 0) && (yy == 0) )
           key = getch();
             else
                key = 1;
             switch(key)
                 {
              /* is it the mouse ? */
```

```
case 1:
  x = x + xx;
       y = y + yy;
       if (y < yymin) y = yymin;
       if (y > ylim) y = ylim;
       if (x < xxmin) x = xxmin;
       if (x > xlim) x=xlim;
       break;
 /* is it a function key ? */
     case 72:                /* UP ARROW */
       y = y - dy;
       if (y < yymin) y = yymin;
     break;
     case 80:                /* DOWN ARROW */
       y = y + dy;
       if (y > ylim) y = ylim;
     break;
     case 75:                /* LEFT ARROW */
       x = x - dx;
       if (x < xxmin) x = xxmin;
     break;
     case 77:                /* RIGHT ARROW */
       x = x + dx;
       if (x > xlim) x=xlim;
     break;
   } /* SWITCH(KEY) INSIDE LOOP */
   if ( (x != ox) || (y != oy) ) {
_setwritemode(_GXOR);
      _moveto(xxmin,oy);
      _lineto(xlim,oy);
      _moveto(ox,yymin);
      _lineto(ox,ylim);
      _moveto(xxmin,y);
      _lineto(xlim,y);
      _moveto(x,yymin);
      _lineto(x,ylim);
      ox = x; oy = y;
   }
break;
case 'e':
case 'E':
    if ( corner_num == corner_lim )
  done2 = TRUE;
break;
case 'q':
case 'Q':
    { clear_bottom(); return; }
break;
case 'd':
case 'D':
    if (corner_num == corner_lim) {
 _settextcolor(0);
    _settextposition(37,0);
    _outtext("                    ");
```

```
                _settextcolor(255);
                    }
                    if (corner_num > 0) corner_num--;
              break;
            } /* SWITCH (KEY) */
      } /* If Key */

         bn = CheckPosition(&bx, &by);
/*
_settextposition(37,0);
sprintf(buf,"Mouse Check: Bn = %d   corner_num = %d  corner_lim = %d   prev_bn =
%d",bn,corner_num,corner_lim,prev_bn);
_outtext(buf);
*/

         /* In the the right ? */
         if ( (bn == RIGHT) && (corner_num < corner_lim) && (prev_bn == 0) ) {
            _setcolor( 255 );
            if ( (xvals[corner_num] != 0.0) || (yvals[corner_num] != 0.0) ) {
                    _setpixel( (short) xvals[corner_num],
                              (short) (2*yvals[corner_num]) );
                    _setpixel( (short) xvals[corner_num],
                              (short) (2*yvals[corner_num]+1) );
            }
            _setpixel( x, y);
            _setpixel( x, y+1);
            xvals[corner_num] = (float) x;
            yvals[corner_num] = (float) y/2;
            switch(corner_num++) {
            case 0:
                    _settextposition(35,0);
                    sprintf(buf,"Left: (%d,%d)",x,y/2);
                    _outtext(buf);
                    break;
            case 1:
                    _settextposition(35,40);
                    sprintf(buf,"Right: (%d,%d)",x,y/2);
                    _outtext(buf);
                    break;
            } /* Switch */
         } /* if bn == right */
         prev_bn = bn;

   } while (!done2);

   NumY = 32;

   num_anal_pts = 0;

     xstart = (float) round ( xvals[0] );
     ystart = (float) round ( yvals[0] );

     xend = (float) round ( xvals[1] );
     yend = (float) round ( yvals[1] );
```

```
    for (j=0; j<=NumY; j++) {

          anal_ptsx[num_anal_pts]  = round( xstart + (xend - xstart) * j / NumY );
          anal_ptsy[num_anal_pts]  = round( ystart + (yend - ystart) * j / NumY );
          anal_sizex[num_anal_pts] = 0;
          anal_sizey[num_anal_pts] = 0;
          anal_type[num_anal_pts] = 0;
          num_anal_pts++;

    } /* j */

   clear_bottom();
   grid_num = LINE_ARRAY;

}



/* ********************************************************************** */
void make_area_grid(void)
{

char buf[80];
int done2, i, j;
int corner_num, corner_lim;
float xvals[4], yvals[4];
float xstart, xend, ystart, yend, NumX, NumY;

  done2 = FALSE;

  clear_bottom();
  _settextcolor(255);
  _settextposition(33,0);
  _outtext("Area Grid Mode              \n");
  _settextposition(34,0);
  _outtext("(E)xit, (D)elete Point, (Q)uit.   RightMouse to Choose");
  _settextposition(35,0);
  _outtext("Upper Left:");
  corner_num = 0;
  corner_lim = 4;
  prev_bn = RIGHT;
  _setwritemode(_GXOR);
  xvals[1] = xvals[2] = xvals[3] = xvals[0] = 0.0;
  yvals[1] = yvals[2] = yvals[3] = yvals[0] = 0.0;

  do {
         switch (corner_num) {
            case 0:
                   _settextposition(35,0);
                   _outtext("Upper Left:              ");
                   break;
            case 1:
                   _settextposition(35,40);
                   _outtext("Upper Right:             ");
```

```
            break;
        case 2:
            _settextposition(36,40);
            _outtext("Lower Right:              ");
            break;
        case 3:
            _settextposition(36,0);
            _outtext("Lower Left:              ");
            break;
        case 4:
            _settextposition(37,0);
            _outtext("Ready to (E)xit              ");
            break;
    } /* Switch corner_num */


key = kbhit();
    GetMickeys(&xx, &yy);

if ( (key != 0) || (xx != 0) || (yy != 0) ) {
      if ( (xx == 0) && (yy == 0) )
    key = getch();
      else
        key = 1;
      switch(key)
      {
        case 1:
        case 0:
        case 224:
          if ( (xx == 0) && (yy == 0) )
        key = getch();
          else
            key = 1;
          switch(key)
              {
            /* is it the mouse ? */
            case 1:
              x = x + xx;
                  y = y + yy;
                  if (y < yymin) y = yymin;
                  if (y > ylim) y = ylim;
                  if (x < xxmin) x = xxmin;
                  if (x > xlim) x=xlim;
                  break;
             /* is it a function key ? */
                case 72:                    /* UP ARROW */
                  y = y - dy;
                  if (y < yymin) y = yymin;
                break;
                case 80:                    /* DOWN ARROW */
                  y = y + dy;
                  if (y > ylim) y = ylim;
                break;
                case 75:                    /* LEFT ARROW */
```

```
                x = x - dx;
                if (x < xxmin) x = xxmin;
              break;
              case 77:                    /* RIGHT ARROW */
                x = x + dx;
                if (x > xlim) x=xlim;
              break;
             } /* SWITCH(KEY) INSIDE LOOP */
             if ( (x != ox) || (y != oy) ) {
         _setwritemode(_GXOR);
               _moveto(xxmin,oy);
               _lineto(xlim,oy);
               _moveto(ox,yymin);
               _lineto(ox,ylim);
               _moveto(xxmin,y);
               _lineto(xlim,y);
               _moveto(x,yymin);
               _lineto(x,ylim);
               ox = x; oy = y;
             }
         break;
         case 'e':
         case 'E':
               if ( corner_num == corner_lim )
            done2 = TRUE;
         break;
         case 'q':
         case 'Q':
               { clear_bottom(); return; }
         break;
         case 'd':
         case 'D':
              if (corner_num == corner_lim) {
           _settextcolor(0);
              _settextposition(37,0);
              _outtext("                    ");
           _settextcolor(255);
              }
              if (corner_num > 0) corner_num--;
         break;
        } /* SWITCH (KEY) */
   } /* If Key */

      bn = CheckPosition(&bx, &by);
/*
_settextposition(37,0);
sprintf(buf,"Mouse Check: Bn = %d  corner_num = %d  corner_lim = %d   prev_bn =
%d",bn,corner_num,corner_lim,prev_bn);
_outtext(buf);
*/

      /* In the the right ? */
      if ( (bn == RIGHT) && (corner_num < corner_lim) && (prev_bn == 0) ) {
         _setcolor( 255 );
```

```
            if ( (xvals[corner_num] != 0.0) || (yvals[corner_num] != 0.0) ) {
                _setpixel( (short) xvals[corner_num],
                        (short) (2*yvals[corner_num]) );
                _setpixel( (short) xvals[corner_num],
                        (short) (2*yvals[corner_num]+1) );
            }
            _setpixel( x, y);
            _setpixel( x, y+1);
            xvals[corner_num] = (float) x;
            yvals[corner_num] = (float) y/2;
            switch(corner_num++) {
            case 0:
                _settextposition(35,0);
                sprintf(buf,"Upper Left: (%d,%d)",x,y/2);
                _outtext(buf);
                break;
            case 1:
                _settextposition(35,40);
                sprintf(buf,"Upper Right: (%d,%d)",x,y/2);
                _outtext(buf);
                break;
            case 2:
                _settextposition(36,40);
                sprintf(buf,"Lower Right: (%d,%d)",x,y/2);
                _outtext(buf);
                break;
            case 3:
                _settextposition(36,0);
                sprintf(buf,"Lower Left: (%d,%d)",x,y/2);
                _outtext(buf);
                break;
            } /* Switch */
        } /* if bn == right */
        prev_bn = bn;

} while (!done2);

NumX = 32;
NumY = 20;

num_anal_pts = 0;

for (i=0; i<=NumX; i++) {

  xstart = (float) round ( xvals[0] + (xvals[1] - xvals[0]) / NumX * i ) ;
  ystart = (float) round ( yvals[0] + (yvals[1] - yvals[0]) / NumX * i );

  xend = (float) round ( xvals[3] + (xvals[2] - xvals[3]) / NumX * i );
  yend = (float) round ( yvals[3] + (yvals[2] - yvals[3]) / NumX * i );

  for (j=0; j<=NumY; j++) {

        anal_ptsx[num_anal_pts]  = round( xstart + (xend - xstart) * j / NumY );
        anal_ptsy[num_anal_pts]  = round( ystart + (yend - ystart) * j / NumY );
```

```
        anal_sizex[num_anal_pts] = 0;
        anal_sizey[num_anal_pts] = 0;
        anal_type[num_anal_pts] = 0;
        num_anal_pts++;

    } /* j */
  } /* i */
  clear_bottom();
  grid_num = RECT_AREA_ARRAY;

}


/* ************************************************************************

        Find the Center of the Wafer as Well as
        the distances involved.
*/
void find_center(centered)
int centered;

{
int xx,yy;
int horiz, vert, done2;
float mv, mh, Lact, CCDx, CCDy, dx, dy, theta, LLx, LLy;

// Save Values
  horiz = vert = FALSE;
  done2 = FALSE;
  lx = ly = rx = ry = tx = ty = box = boy = 0;
  Lact = 8.0e4;   /* uM */

  clear_bottom();
  _settextcolor(255);
  _settextposition(33,0);
  _outtext("Center Find Mode                \n");
  _settextposition(34,0);
  _outtext("(E)xit when Done  (D)ie Scaling, (P)attern Scale, (W)afer Scaling, (Q)uit");
  _settextposition(35,0);
  _outtext("Left Mouse for Horiz. Line, Right Mouse for Vert Line");



  do {

     key = kbhit();
          GetMickeys(&xx, &yy);

     if ( (key != 0) || (xx != 0) || (yy != 0) ) {
          if ( (xx == 0) && (yy == 0) )
          key = getch();
            else
              key = 1;
            switch(key)
```

```
{
  case 1:
  case 0:
  case 224:
   if ( (xx == 0) && (yy == 0) )
  key = getch();
   else
      key = 1;
   switch(key)
        {
     /* is it the mouse ? */
     case 1:
       x = x + xx;
            y = y + yy;
            if (y < yymin) y = yymin;
            if (y > ylim) y = ylim;
            if (x < xxmin) x = xxmin;
            if (x > xlim) x=xlim;
            break;
      /* is it a function key ? */
          case 72:                      /* UP ARROW */
             y = y - dy;
             if (y < yymin) y = yymin;
          break;
          case 80:                      /* DOWN ARROW */
             y = y + dy;
             if (y > ylim) y = ylim;
          break;
          case 75:                      /* LEFT ARROW */
             x = x - dx;
             if (x < xxmin) x = xxmin;
          break;
          case 77:                      /* RIGHT ARROW */
             x = x + dx;
             if (x > xlim) x=xlim;
          break;
        } /* SWITCH(KEY) INSIDE LOOP */
        if ( (x != ox) || (y != oy) ) {
  _setwritemode(_GXOR);
            _moveto(xxmin,oy);
            _lineto(xlim,oy);
            _moveto(ox,yymin);
            _lineto(ox,ylim);
            _moveto(xxmin,y);
            _lineto(xlim,y);
            _moveto(x,yymin);
            _lineto(x,ylim);
            ox = x; oy = y;
        }
  break;
  case 'e':
  case 'E':
        if ( (horiz == TRUE) && (vert == TRUE) )
     done2 = TRUE;
```

```
            break;
            case 'd' :
            case 'D' :
                 Lact = 1.0e4;      /* uM */
            break;
            case 'p' :
            case 'P' :
                 Lact = 8.0e4;      /* uM */
            break;
            case 'w' :
            case 'W' :
                 Lact = 10.0e4;     /* uM */
            break;
            case 'q' :
            case 'Q' :
                 { clear_bottom(); return; }
            break;
          } /* SWITCH (KEY) */
} /* If Key */

     bn = CheckPosition(&bx, &by);

     /* In the the Left ? */
     if ( (bn == LEFT) && (prev_bn == LEFT) ) {
       if ( (prev_bx != bx) && (prev_by != by) ) {
    _setwritemode(_GXOR);
          _moveto( (short) prev_bx, (short) prev_by);
          _lineto( (short) x, (short) y);
          _moveto( (short) prev_bx, (short) prev_by);
          _lineto( (short) x, (short) y);
       }
     }

     /* Wait until a button is pressed and mark its location */
     if ( (bn == LEFT) && (prev_bn == 0) ) {
        prev_bn = bn;
        prev_bx = x;
        prev_by = y;
        /* If a line is there, erase it */
        if ( (lx != 0) && (rx != 0) && (ly != 0) && (ry != 0) ) {
      _setwritemode(_GXOR);
      _moveto( (short) lx, (short) ly);
      _lineto( (short) rx, (short) ry);
        }
     }

     /* Was the button just released ??? */
     if ( (bn == 0) && (prev_bn == LEFT) ) {
       _moveto( (short) prev_bx, (short) prev_by);
       _lineto( (short) x, (short) y);
       lx = prev_bx;
       ly = prev_by;
       rx = x;
       ry = y;
```

```
      horiz = TRUE;
      prev_bn = 0;
      prev_bx = 0;
      prev_by = 0;
    }

    /* In the middle ? */
    if ( (bn == RIGHT) && (prev_bn == RIGHT) ) {
      if ( (prev_bx != bx) && (prev_by != by) ) {
    _setwritemode(_GXOR);
        _moveto( (short) prev_bx, (short) prev_by);
        _lineto( (short) x, (short) y);
        _moveto( (short) prev_bx, (short) prev_by);
        _lineto( (short) x, (short) y);
      }
    }

    /* Wait until a button is pressed and mark its location */
    if ( (bn == RIGHT) && (prev_bn == 0) ) {
       prev_bn = bn;
       prev_bx = x;
       prev_by = y;
       if ( (tx != 0) && (box != 0) && (ty != 0) && (boy != 0) ) {
      _setwritemode(_GXOR);
      _moveto( (short) tx, (short) ty);
      _lineto( (short) box, (short) boy);
       }
    }

    /* Was the button just released ??? */
    if ( (bn == 0) && (prev_bn == RIGHT) ) {
       _moveto( (short) prev_bx, (short) prev_by);
       _lineto( (short) x, (short) y);
       tx = prev_bx;
       ty = prev_by;
       box = x;
       boy = y;
       vert = TRUE;
       prev_bn = 0;
       prev_bx = 0;
       prev_by = 0;
    }

  } while (!done2);

// Erase the lines that were drawn.

  _moveto( (short) lx, (short) ly);
  _lineto( (short) rx, (short) ry);
  _moveto( (short) tx, (short) ty);
  _lineto( (short) box, (short) boy);

// Find the Center from the intersection of 2 lines.
```

```
   ly = ly /2; ry = ry / 2; ty = ty / 2; boy = boy/2;
   mh = (float) (ry-ly) / (float) (rx-lx);
   mv = (float) (ty-boy) / (float) (tx-box);
   centx = ((float)ty - (float)ly + mh*(float)lx - mv*(float)tx )/(mh-mv);
   centy = (float) ly + mh * (centx - lx);

// Now, Calculate the size of a pixle on the actual surface.

   CCDx = 11.5;            /* uM */
   CCDy = 27.0;
//  Lact = 8.0e4;  /* uM */

   dx = (float) abs(rx-lx);
   dy = (float) abs(ry-ly);

   theta = atan2( (double) (dy*CCDy), (double) (dx*CCDx));
//  theta2= atan2( (double) (ry-ly), (double) (rx-lx));
   theta2= theta;
   LLx  = Lact * cos( (double) theta);
   LLy  = Lact * sin( (double) theta);

   mapx = LLx/dx;
   mapy = LLy/dy;

   dx = (float) (rx-lx);
   dy = (float) (ry-ly);
   theta2 = atan2( (double) (dy*CCDy), (double) (dx*CCDx));

   LLx  = Lact * cos( (double) theta);
   LLy  = Lact * sin( (double) theta);

   mapx = LLx/dx;
   mapy = LLy/dy;

   dx = (float) abs(rx-lx);
   dy = (float) abs(ry-ly);
   centered = TRUE;

   centered = TRUE;

   clear_bottom();

}

/* ************************************************************************* */

static float sqrarg;
// #define SQR(a) (sqrarg=(a),sqrarg*sqrarg)
#define SQR(a) ((a) * (a))

void fit(x,y,ndata,sig,mwt,a,b,siga,sigb,chi2,q)
float x[], y[], sig[], *a, *b, *siga, *sigb, *chi2, *q;
int ndata, mwt;
```

```
/*
given a set of points x[1..ndata], y[1..ndata] with standard
deviations sig[1..ndata], fit them to a straight line y=a+bx
by minimizing chi-squared. Returned are a and b and their
respective probable uncertainties siga and sigb, the chi-
squared chi2, and the goodness-of-fit probability q (that the
fit would have chi-squared this large or larger). If mwt = 0
on input, then the standard deviations are assumed to be
unavailable: q is returned as 1.0 and the normalization
of chi2 is to unit standard deviation on all points.

*/

{
  int i;
  float wt, t, sxoss, sx=0.0, sy=0.0, st2=0.0, ss, sigdat;

/* gammq is commented out as it is only needed to calculate
  q for cases where the weights are known.

  float gammq();
*/

  *b = 0.0;
  /* Accumulate Sums */
  if (mwt) {
    /* with weights */
    ss=0.0;
    for (i=1; i<=ndata;i++) {
      wt = 1.0/SQR(sig[i]);
      ss += wt;
      sx += x[i] * wt;
      sy += y[i] * wt;
    } /* for */
  } else {
    /* or without weights */
    for (i=1; i<=ndata; i++) {
      sx += x[i];
      sy += y[i];
          } /* for */
    ss = ndata;
  } /* if mwt */

  sxoss = sx/ss;

  if (mwt) {
    for (i=1; i<=ndata; i++) {
      t = (x[i]-sxoss)/sig[i];
      st2 += t*t;
      *b += t*y[i]/sig[i];
    } /* for */
  } else {
    for (i=1; i<=ndata; i++) {
      t = x[i]-sxoss;
```

```
      st2 += t*t;
      *b += t*y[i];
    } /* for */
  } /* if mwt */

  /* solve for a,b, siga and sigb */
  *b /= st2;
  *a = (sy-sx*(*b))/ss;
  *siga = sqrt((1.0+sx*sx/(ss*st2))/ss);
  *sigb = sqrt(1.0/st2);

  *chi2 = 0.0;

/* Calculate Chi Squared */

/* for unweighted data, evaluate typical sig using chi2
  and adjust the standard deviations */

  if (mwt == 0) {
    for (i=1; i<=ndata; i++)
      *chi2 += SQR(y[i]-(*a)-(*b)*x[i]);
    *q = 1.0;
    sigdat = sqrt( (*chi2)/(ndata-2));
    *siga *= sigdat;
    *sigb *= sigdat;
  }

  else {
    for (i=1; i<=ndata; i++)
      *chi2 += SQR((y[i]-(*a)-(*b)*x[i])/sig[i]);
//    *q = gammq(0.5*(ndata-2), 0.5*(*chi2));
  }

} /* FIT */



/* ************************************************************************* */

void four1(data, nn, isign)
float data[];
int nn, isign;

/* replaces data by its discrete fourier transform, if isign
  is input as 1; or replaces data by nn times its inverse
  discrete fourier transform, if isign is input as -1.  data
  is a complex array of length nn, input as a real array
  data[1..2*nn].  nn MUST be an integer power of 2.
  This is NOT checked for.
*/

{
  int n, mmax, m, j, istep, i;
  /* double precision for the trigonometric recurrences */
```

```
double wtemp, wr, wpr, wpi, wi, theta;
float tempr, tempi;

n = 2 * nn;
j = 1;
/* do the bit reversal */
for (i=1; i<=n; i+=2) {
  if (j>i) {
    /* Exchnge a complex number */
        tempr = data[j];
        tempi = data[j+1];
        data[j] = data[i];
        data[j+1] = data[i+1];
        data[i] = tempr;
        data[i+1] = tempi;
  }
  m = n / 2;
  while ((m >= 2) && (j > m)) {
    j -= m;
    m = m / 2;
  }
  j += m;
} /* for */

/* begin the Danielson-Lanczos Routine */
mmax = 2;
/* outer loop executes log2(nn) times */
while (n > mmax) {
  istep = 2 * mmax;
  /* initialize for the triginometric reurrences */
  theta = 6.28318530717959/(isign*mmax);
  wtemp = sin(0.5 * theta);
  wpr = -2.0 * wtemp * wtemp;
  wpi = sin(theta);
  wr = 1.0;
  wi = 0.0;
  for (m=1; m<=mmax; m+= 2) {
    for (i=m; i<= n; i += istep) {
      j = i + mmax;
      tempr = wr*data[j]-wi * data[j+1];
      tempi = wr*data[j+1] + wi*data[j];
      data[j] = data[i] - tempr;
      data[j+1] = data[i+1]-tempi;
      data[i] += tempr;
      data[i+1] += tempi;
    }
        wtemp = wr;
    wr = wr*wpr - wi*wpi+wr;
    wi = wi*wpr + wtemp*wpi+wi;
  } /* FOR */
  mmax = istep;
 } /* WHILE */
} /* FOUR1 */
```

```
/* ************************************************************************ */
void draw_y_axis(start, end, step)
double start, end, step;
{
#define YTICKSIZE 0.020
  double y;
  char buf[40];

  y = start;
  while (y <= end) {
    _moveto_w(0.0, y);
    _lineto_w(-YTICKSIZE*(xmax-xmin), y);
    sprintf(buf,"%4.1f",y);
    _moveto_w(-(xmax-xmin)*0.1, y);
    _outgtext(buf);
    y += step;
  }

}

/* ************************************************************************ */
void draw_x_axis(start, end, step)
double start, end, step;
{
#define XTICKSIZE 0.035
  double x;
  char buf[40];

  x = start;
  while (x <= end) {
    _moveto_w(x, ymin);
    if (ymin == 0.0)
       _lineto_w(x, -ymax * XTICKSIZE);
    else
       _lineto_w(x, ymin - (ymax-ymin) * XTICKSIZE);
    sprintf(buf,"%1.2f",x);
    _moveto_w(x - (end-start)*0.02, ymin - (ymax-ymin) * 1.5 * XTICKSIZE);
    _outgtext(buf);
    x += step;
  }

}
/* ************************************************************************ */

void plot_signal(void)
{
  int k,j;
  for (j=1; j<=numpts; j++)
  {
    _setcolor(j);
    k = start;
    _moveto_w( t[k], yv[k][j]);
    for (k=start+1; k<=cnt; k++)
        _lineto_w( t[k], yv[k][j]);
```

```
  }

} /* PLOT_SIGNAL */


/* ************************************************************************* */
void calc_er(j, er, n)
int j;
float *er, n;
{
  float delta;
  delta = 753.4 / 2.0 / n;    /* Distance Of a Fringe in nm */
  *er = delta / sdelta * fmax[j] * 600.0;       /* A/Min */
}

/* ************************************************************************* */
void plot_er()
{
int j;
int xl,yt,xr,yb;
float er1, er2, er3;
char buf[80];


   if (_registerfonts("\\c700\\bin\\*.fon") <= 0)
   {
      _setvideomode( _DEFAULTMODE );
          printf("UNABLE TO REGISTER FONTS\n");
          exit(1);
   }
   if (_setfont("t'tms rmn'h30w24b") <0 )
   {
      _setvideomode( _DEFAULTMODE );
          printf("UNABLE TO SET FONT\n");
          exit(1);
   }
   _getfontinfo( &fi);

  for (j=0; j<num_anal_pts; j++)
  {
    xl =  anal_ptsx[j] - anal_sizex[j];
    xr =  anal_ptsx[j] + anal_sizex[j];
    yt = (anal_ptsy[j] - anal_sizey[j]) * 2;
    yb = (anal_ptsy[j] + anal_sizey[j]) * 2;

    _setwritemode(_GPSET);
    _setcolor(255);
    _moveto(xl,yb);
    _lineto(xl,yt);
    _lineto(xr,yt);
    _setcolor(0);
    _lineto(xr,yb);
    _lineto(xl,yb);
```

```
    calc_er(j+1, &er1, 1.50);          /* Photoresist */
    calc_er(j+1, &er2, 1.42);          /* Oxide */
    calc_er(j+1, &er3, 3.42);          /* Poly */
    sprintf(buf,"%5.0f,%5.0f,%5.0f",er3,er2,er1);
    _moveto( anal_ptsx[j], anal_ptsy[j]*2);
    _outgtext(buf);

  } /* FOR */
}

/* ********************************************************************** */
void plot_fft(void)
{
  int k,j;
  float er1, er2, er3, xx, yy;
  char buf[80];
  for (j=1; j<=numpts; j++)
  {
    _setcolor(j);
    k = 1;
    _moveto_w( fabs(freq[k]), fmag[k][j]);
    for (k=2; k<=pow2/2+1; k++)
        _lineto_w( fabs(freq[k]), fmag[k][j]);

    _moveto_w( fabs( freq[ fmaxi[j] ]), 0.0);
    _lineto_w( fabs( freq[ fmaxi[j] ]), fmag[ fmaxi[j]][j]);
    calc_er(j, &er1, 1.50); /* Photoresist */
    calc_er(j, &er2, 1.42); /* Oxide */
    calc_er(j, &er3, 3.42); /* Poly */
    xx = r[j] * cos( th[j]*pi/180. + theta);
    yy = r[j] * sin( th[j]*pi/180. + theta);
    sprintf(buf,"%5.0f  %5.0f  %5.0f (%2.2f,%2.2f)cm ",er3,er2,er1,xx,yy);
    _moveto_w((xmax-xmin)*0.5,ymax-(j+3)*ymax*.045);
    _outgtext(buf);
  }

} /* PLOT_FFT */

/* ********************************************************************** */
void plot(yymax,xxmax)
float yymax,xxmax;
{
  int i;
  int xleft, xright, ytop, ybottom;

//  mode= _XRES16COLOR;                /* 1024 x 768 x 16 */
  mode= _SRES16COLOR;
  _setvideomode( mode );
  _getvideoconfig( &vc);

  if (_registerfonts("\\c700\\bin\\*.fon") <= 0)
  {
    _setvideomode( _DEFAULTMODE );
       printf("UNABLE TO REGISTER FONTS\n");
```

```
        exit(1);
    }
    if (_setfont("t'tms rmn'h30w24b") <0 )
    {
      _setvideomode( _DEFAULTMODE );
        printf("UNABLE TO SET FONT\n");
        exit(1);
    }
    _getfontinfo( &fi);

    /* Make a Viewport That is 1/2 the screen in Height, Full Width */
    xleft = 0;
    xright = vc.numxpixels;
    ytop = 0;
    ybottom = vc.numypixels/2;

    _setviewport(xleft,ytop,xright,ybottom);

    ymax = yymax;
    ymin = 0.0;
    xmax = 0.15;
    xmin = 0.0;

    _setwindow(TRUE, -(xmax-xmin)*0.2, ymax*1.1, (xmax-xmin)*1.025, -0.2*ymax);
    _setcolor((short)_WHITE);
    _rectangle_w( _GBORDER, -(xmax-xmin)*0.2, ymax*1.1,
      (xmax-xmin)*1.025, -0.2*ymax);
    _rectangle_w( _GBORDER, xmin, ymax, xmax, ymin);

    draw_x_axis(xmin, xmax, (xmax-xmin)/5.0);
    _moveto_w((xmax-xmin)*0.38,-0.13*ymax);
    _outgtext("Frequency * Sampling Rate");

    draw_y_axis(ymin, ymax, ymax/5.0);
    _setgtextvector(0,1);
    _moveto_w((xmax-xmin)*-.18, .25*ymax);
    _outgtext("FFT Magnitude");
    _setgtextvector(1,0);

    plot_fft();
    if (_setfont("t'tms rmn'h80w60b") <0 )
    {
      _setvideomode( _DEFAULTMODE );
        printf("UNABLE TO SET FONT\n");
        exit(1);
    }
    _setcolor((short)_WHITE);
    _moveto_w((xmax-xmin)*.4,1.05*ymax);
    _outgtext("FOURIER TRANSFORM");

    _moveto_w((xmax-xmin)*0.5,ymax-0.5*ymax*.045);
    _outgtext(" ETC: ;RING RATE A/Min");
    _moveto_w((xmax-xmin)*0.5,ymax-2*ymax*.045);
    _outgtext(" POLY   Oxide   Resist   Position");
```

```
/* AND NOW THE SECOND GRAPH */
_setviewport(xleft,ybottom,xright,vc.numypixels);
ymax = 1.5;
ymin = -1.5;
xmax = xxmax;
xmin = 0.0;

_setwindow(TRUE, -(xmax-xmin)*0.2, ymax*1.2, (xmax-xmin)*1.045,
 1.4*ymin);
_setcolor((short)_WHITE);
_rectangle_w( _GBORDER, -(xmax-xmin)*0.2, ymax*1.2,
   (xmax-xmin)*1.045, 1.4*ymin);
_rectangle_w( _GBORDER, xmin, ymax, xmax, ymin);

draw_x_axis(xmin, xmax, (xmax-xmin)/5.0);
_moveto_w((xmax-xmin)*0.38, ymin*1.3);
_outgtext("Time (Seconds)");
_moveto_w((xmax-xmin)*0.7, ymin*1.3);
_outgtext("PRESS ANY KEY, (Q) TO QUIT");

draw_y_axis(ymin, ymax, (ymax-ymin)/6.0);
_setgtextvector(0,1);
_moveto_w((xmax-xmin)*-.18, ymin + .25*(ymax-ymin));
_outgtext("CCD Signal Intensity");
_setgtextvector(1,0);

plot_signal();
if (_setfont("t'tms rmn'h80w60b") <0 )
{
  _setvideomode( _DEFAULTMODE );
      printf("UNABLE TO SET FONT\n");
      exit(1);
}
_setcolor((short)_WHITE);
_moveto_w((xmax-xmin)*.4,1.10*ymax);
_outgtext("CCD Signal");


}


/* ***********************************************************************

        CCD Data Analysis.
        Linear Fit to remove DC Components and
        FFT of Data

        TJD, MIT Dept. of Chemical Engineering
        05/26/92
*/
void analyze(void)
{
  int     i,j,k,keyhit,resp,loc_found,nnmin;
```

```
FILE            *apts, *dsum;
float  a,b,aa,bb,xx,yy,tlim;        /* Fit Values */

_setvideomode( _DEFAULTMODE);
printf("Beginning Execution\n");

apts = fopen("analpts.dat", "r");
if (apts == NULL) {
       printf("Error Opening ANALPTS.DAT\n");
       exit(1);
}
fscanf(apts,"%d",&numpts);
if (numpts >= NMAX) {
       printf("numpts > nmax!\n");
       exit(1);
}
for (i=1; i<= numpts; i++)
       fscanf(apts,"%d %d %d %d %d",&j,&j,&j,&j,&j);
fscanf(apts, "%d %d", &centx, &centy);
fscanf(apts, "%f %f %f", &xpix, &ypix, &theta);
fclose(apts);

dsum = fopen("dsum.dat", "r");
 if (dsum == NULL) {
       printf("Error Opening DSUM.DAT\n");
       exit(1);
}

for (i=1; i<=numpts; i++) {
  fscanf(dsum, "%f", &(r[i]) );
} /* FOR i*/

for (i=1; i<=numpts; i++) {
  fscanf(dsum, "%f", &(th[i]) );
} /* FOR i*/

start = -1;
cnt = 1;
while (!feof(dsum)) {
   fscanf(dsum, "%f", &(t[cnt]) );
       if (!feof(dsum)) {
     for (i=1; i<=numpts; i++) {
       fscanf(dsum, "%f", &(yv[cnt][i]) );
     } /* FOR i*/
         if ( (start == -1) && (yv[cnt][1] != 0.0)) start = cnt;
     cnt++;
       } /* !feof(dsum) */
} /* WHILE */
cnt--;
printf("Finished Reading in %d Data Points\n",cnt);
printf("First Nonzero Data Point is %d\n",start);
printf("\n\nNonZero Data covers [%f,%f] Seconds.\n\n",t[start],t[cnt]);

fclose(dsum);
```

```
printf("Modify Start/End Times for FFT Analysis ? (Y/N) :");
do {
  keyhit = kbhit();
  if ( (keyhit=kbhit() ) != 0) keyhit = getch();
}
while (!(keyhit == 'y' || keyhit == 'Y' || keyhit == 'n' || keyhit == 'N'));
printf("%c\n\n",keyhit);

resp = keyhit;

if ((resp=='y') || (resp=='Y'))
  {
        printf("Start Time = %f\n", t[start]);
        printf("Enter Time to Start at: ");
        scanf("%f",&tlim);
        i = start - 1;
        loc_found = FALSE;
        do {
          i++;
          if ( (t[i] >= tlim) || (i >= cnt) ) {
                    start = i;
                    loc_found = TRUE;
          } }
        while (!loc_found);
        printf("Start Time found at point %d with Time = %f\n",start,t[start]);

        printf("End Time = %f\n", t[cnt]);
        printf("Enter Time to End at: ");
        scanf("%f",&tlim);
        i = start - 1;
        loc_found = FALSE;
        do {
          i++;
          if ( (t[i] >= tlim) || (i >= cnt) ) {
                    cnt = i;
                    loc_found = TRUE;
          } }
        while (!loc_found);
        printf("End Time found at point %d with Time = %f\n",cnt,t[cnt]);
  }


nn = 0;
sum = 0.0;
/* Start At the Second Data Point, zero or non-zero */
for (i=2; i<=cnt; i++) {
       nn++;
       sum += (t[i]-t[i-1]);
}
sdelta = sum / (float)nn;

printf("Sampling Rate = %f Hz,   Sampling Time = %f sec.\n",
1.0/sdelta,sdelta);
```

```
/* Now, Figure Out The Power of Two! */
actpts = cnt - start +1;
nnmin = nn = (int) ceil( log( (double) actpts) / log(2.0));
printf("For %d Points, Nearest Power of 2 Above is %d\n",actpts,nn);
nn += 3;
if (nn > NNMAX) nn = NNMAX;
printf("Recommended Transform is 2^%d\n\n",nn);

printf("Modify Transform Size ? (Y/N) :");
do {
 keyhit = kbhit();
 if ( (keyhit=kbhit() ) != 0) keyhit = getch();
}
while (!(keyhit == 'y' || keyhit == 'Y' || keyhit == 'n' || keyhit == 'N'));
printf("%c\n\n",keyhit);

resp = keyhit;

if ((resp=='y') || (resp=='Y'))
  {
        printf("Enter Transform Size, NN <= %d :   ",NNMAX);
        scanf("%d",&nn);
        if ( (nn < nnmin) || (nn > NNMAX) ) nn = NNMAX;
  }

fpow2 = pow(2.0,(double)nn);
pow2  = (int) fpow2;

printf("Using a FFT with N = %d -> 2**N = %d\n",nn,pow2);

/* Calculate The Frequencies */
j = 1;
for (i=pow2+1; i<=2*pow2; i+=2) {
      freq[j] = (float) ( (i - (2*pow2+1) )/2) / fpow2;
      j++;
}
for (i=1; i<=pow2+1; i+= 2) {
      freq[j] = (float) ( (i-1)/2 ) / fpow2;
      j++;
}


/* DO A BASELINE SUBTRACTION */

for (j=1; j<=numpts; j++) {

      /* Transfer Data to a Temporary Array */
      for (i=start; i<=cnt; i++)
        ycall[i] = yv[i][j];

      /* Fit A Straight Line */
      fit( &(t[start-1]), &(ycall[start-1]), actpts-1, &(dum[0]), 0,
          &a1, &b1, &siga, &sigb, &chi2, &q);
```

```
/* Subtract off Baseline, And Find Maximum Value */
ym[j] = -999.9;
for (i=start; i<=cnt; i++) {
   yv[i][j] -= (a1+b1*t[i]);
   if (yv[i][j] > ym[j]) ym[j] = yv[i][j];
} /* FOR I*/

for (i=start; i<=cnt; i++)
   yv[i][j] /= ym[j];

/* TAKE THE FFT */
for (k=1; k<=2*pow2+1; k++)
   fft[k] = 0.0;

/* Transfer The Data */
k = 1;
for (i=start; i<=cnt; i++) {
   fft[k]   = yv[i][j];
   fft[k+1] = 0.0;
   k += 2;
}

/* Zero The First Five Points too! */
k = 1;
for(i=start; i<=(start+5); i++) {
   fft[k]   = 0.0;
   fft[k+1] = 0.0;
   k += 2;
}

/* Calculate The FFT */
four1(fft, pow2, 1);


/* Check 1/2 The Spectrum for the Dominant Frequency Component */
k = 1;
fmaxi[j] = 0;
sum = 0.0;
for (i=pow2+1; i<=2*pow2; i+=2) {
   fmag[k][j] = sqrt( fft[i]*fft[i] + fft[i+1]*fft[i+1] );
   if (fmag[k][j] > sum) {
         sum = fmag[k][j];
         fmaxi[j] = k;
   }
   k++;
}
for (i=1; i<=pow2+1; i+=2) {
   fmag[k][j] = sqrt( fft[i]*fft[i] + fft[i+1]*fft[i+1] );
   k++;
}

/* Now, we have a candidate for the maximum, lets find it! */
ycall[1] = fmag[ fmaxi[j]-2][j];
ycall[2] = fmag[ fmaxi[j]-1][j];
```

```
            ycall[3] = fmag[ fmaxi[j]  ][j];

            /* Should be -2, but 0th elemnt is skipped, used 1 less */
            fit( &(freq[fmaxi[j]-3]), ycall, 3, &(dum[0]), 0,
                &a, &b, &siga, &sigb, &chi2, &q);

            ycall[1] = fmag[ fmaxi[j]  ][j];
            ycall[2] = fmag[ fmaxi[j]+1][j];
            ycall[3] = fmag[ fmaxi[j]+2][j];

            /* Should be freq[fmaxi[j]], but must skip 0th element */
            fit( &(freq[fmaxi[j]-1]), ycall, 3, &(dum[0]), 0,
                &aa, &bb, &siga, &sigb, &chi2, &q);

            fmax[j] = fabs( (double) ((aa-a)/(bb-b)) );
            xx = r[j] * cos( th[j]*pi/180. + theta);
            yy = r[j] * sin( th[j]*pi/180. + theta);
            printf("%f   %f   %f   %d   (%f,%f)\n",
                  r[j],th[j],1./fmax[j],j,xx,yy);

    } /* FOR J */
    sum = 0.0;
    for (j=1; j<= numpts; j++)
            if (fmag[fmaxi[j]][j] > sum) sum = fmag[fmaxi[j]][j];

    printf("\nPress Return to Graph The Results.\n");
    getch();

    plot(sum,t[cnt]);
}

void readcam(void)
{

// Read the camera:

    highcam (base, refresh_flag, ab_flag, interlace_flag, 0,
           exposure_time, buffer);
    if (interlace_flag != 0)
      highcam (base, refresh_flag, ab_flag, interlace_flag, !0,
             exposure_time, buffer2);
}

void main_menu(void)
{
  char buffer[80];

  _settextcolor(255);
  _settextposition(32,0);
  _outtext("Rectangle Draw Mode");
  _settextposition(32,60);
  sprintf(buffer, "num_anal_pts = %d      ",num_anal_pts);
  _outtext(buffer);
  _settextposition(33,60);
```

```
  sprintf(buffer,"Grid Type = %s  ",grid_type[grid_num]);
  _outtext(buffer);
  _settextposition(33,0);
  _outtext("(W) Round Grid, (L)ine, (A)nalyze, (C)ontinue-Restart");
  _settextposition(34,0);
  _outtext("(Q)uit, (D)one-Write Analpts.Dat, (R)eplay & Write dsum.dat");
  _settextposition(35,0);
  _outtext("(g)ray scale, (G)-Color Scale, (h)-2nd Color Scale");
  _settextposition(36,0);
  _outtext("Mouse Center: Find Center, Mouse Right: Square Grid");
  _settextposition(37,0);
  _outtext("Mouse Left: Draw Analysis Boxes, Click & Drag.  (E)rase Last");
}

void _cdecl
main (void)
{

   int TSENG_4000;
   int i,j,saveit;
   char resp;
   int xl,yt,xr,yb;

  pi = 4.0 * atan(1.0);
  if((ii = CheckMouse()) == 0)
    { puts("No mouse found"); exit(-1); }

   printf("Interlaced Mode ? (Y/N) : ");
   scanf("%s",&resp);
   if ((resp=='y') || (resp=='Y'))
     interlace_flag = !0;
   else
     interlace_flag = 0;

   printf("Rotate Rectangles ? (Y/N) : ");
   scanf("%s",&resp);
   if ((resp=='y') || (resp=='Y'))
     rotate = TRUE;
   else
     rotate = FALSE;

   printf("Enter File Path (include all \\ chars) : ");
   scanf("%s",&fpath);
   strcat(fpath, "image%d.dat");
   printf("File Path = %s\n",fpath);

   printf("\nEnter the image number for the initial display: ");
   scanf("%d",&initnum);
   printf("Contrast Enhance Initial Image ? ");
   scanf("%s",&resp);
   if ((resp=='y') || (resp=='Y'))
     contr = TRUE;
   else
     contr = FALSE;
```

```
    printf("\nReady to Process ? (Skip Preproc) (Y/N) : ");
    scanf("%s",&resp);


// Allocate the image buffers:


    for (i=0; i<V; i++) {

      if ((buffer[i] =
         (pixel _far *) calloc ((size_t) H, sizeof (pixel)))
         == (pixel _far *) NULL) {

         cprintf ("\n\n\rCannot allocate memory for buffer!\n\r");
         exit ((char) -1);
      }
    } /* FOR i */

    if (interlace_flag != 0)  {
      for (i=0; i<V; i++) {

         if ((buffer2[i] =
            (pixel _far *) calloc ((size_t) H, sizeof (pixel)))
            == (pixel _far *) NULL) {

            cprintf ("\n\n\rCannot allocate memory for buffer2!\n\r");
            exit ((char) -1);
         }
      } /* FOR i */
    } /* IF */

// Build display buffers from image buffers:

    for (i=0, j=0; i<V; i++, j+=2) {
      display_buffer[j] = buffer[i];
      if (interlace_flag != 0)
            display_buffer[j+1] = buffer2[i];
          else
            display_buffer[j+1] = buffer[i];
    }

    if ( (resp == 'y') || (resp == 'Y') ) {
          /* init 800 x 600 mode */
      if ((TSENG_4000 = init800 ()) != 0) {
            cprintf ("\r\nTSENG 4000 VGA chip set not detected!");
            exit (-1);
          }
          plotinit(initnum);
          resp = getch();
          if ( (resp == 'q') || (resp == 'Q') ) {
            restore_screen();
            exit(0);
          }
          fsum = fopen("dsum.dat","w");
```

```
        calc_radius();
        plotimage();
        fclose(fsum);
        restore_screen();
        exit(0);
}
else {
    mode= _SRES256COLOR;
    _setvideomode( mode );
    _getvideoconfig( &vc);

    filenum = 0;
    saveit = 0;
    gray_scale();
    plotinit2(initnum);

//   getch();
//   getch();
//   gray_scale();
//   color_scale();
//   getch();
//
//   contrast();
//   tim_display_image();
//
//   getch();
//   gray_scale();
//   getch();
//   two_color_scale();

     done = FALSE;
     /* xlim = vc.numxpixels; */
     xlim = 800;
     /* ylim = vc.numypixels; */
     ylim = 488;
     xxmin = 44;
     yymin = 0;
     ox = x = xlim/2;
     oy = y = ylim/2;
     dx = 1;
     dy = 1;
     prev_bn = 0;
     prev_bx = 0;
     prev_by = 0;

//   ReSet the Wafer Analysis Values
          read_anal_points(FALSE);
     num_anal_pts = 0;


          /* Draw Inital Cursor */
     _setwritemode(_GXOR);
     _setcolor( 255 );
     _moveto(xxmin,oy);
```

```
_lineto(xlim,oy);
_moveto(ox,yymin);
_lineto(ox,ylim);

GetMickeys(&xx, &yy);
    main_menu();

do {

 key = kbhit();
     GetMickeys(&xx, &yy);

 if( (key != 0) || (xx != 0) || (yy != 0) ) {
        if( (xx == 0) && (yy == 0) )
     key = getch();
        else
          key = 1;
       switch(key)
       {
         case 1:
         case 0:
         case 224:
          if( (xx == 0) && (yy == 0) )
        key = getch();
          else
             key = 1;
         switch(key)
              {
           /* is it the mouse ? */
           case 1:
            x = x + xx;
                 y = y + yy;
                 if (y < yymin) y = yymin;
                 if (y > ylim) y = ylim;
                 if (x < xxmin) x = xxmin;
                 if (x > xlim) x=xlim;
                 break;
         /* is it a function key ? */
               case 72:                  /* UP ARROW */
                 y = y - dy;
                 if (y < yymin) y = yymin;
               break;
               case 80:                  /* DOWN ARROW */
                 y = y + dy;
                 if (y > ylim) y = ylim;
               break;
               case 75:                  /* LEFT ARROW */
                 x = x - dx;
                 if (x < xmin) x = xmin;
               break;
               case 77:                  /* RIGHT ARROW */
                 x = x + dx;
                 if (x > xlim) x=xlim;
               break;
```

```
            } /* SWITCH(KEY) INSIDE LOOP */
            if( (x != ox) || (y != oy) ) {
              _moveto(xxmin,oy);
              _lineto(xlim,oy);
              _moveto(ox,yymin);
              _lineto(ox,ylim);
              _moveto(xxmin,y);
              _lineto(xlim,y);
              _moveto(x,yymin);
              _lineto(x,ylim);
              ox = x; oy = y;
            }
        break;
        case 'g':
            gray_scale();
        break;
        case 'G':
            two_color_scale();
        break;
        case 'h':
            color_scale();
        break;
        case 'q':
        case 'Q':
        case 27:
          done = TRUE;
        break;
        case 'd':
        case 'D' :
//           Write an output file suitable for Analysis Input.
    _settextcolor(255);
    _settextposition(35,70);
    _outtext("WRITING ANALPTS.DAT");
            fout = fopen("analpts.dat","w");
            fprintf(fout, "%d\n",num_anal_pts);
            for (i=0; i<num_anal_pts; i++) {
              fprintf(fout,"%d %d %d %d  %d\n", anal_ptsx[i], anal_ptsy[i],
              anal_sizex[i], anal_sizey[i], anal_type[i] );
            } /* FOR */
            fprintf(fout, "%d %d\n", centx, centy);
            fprintf(fout, "%f %f %f\n", mapx, mapy, theta2);
            fclose(fout);
    _settextposition(35,70);
    _outtext("                    ");
        break;
        case 'r':
        case 'R':

          _setvideomode( _DEFAULTMODE);

              /* init 800 x 600 mode */
      if ((TSENG_4000 = init800 ()) != 0) {
          cprintf ("\r\nTSENG 4000 VGA chip set not detected!");
          exit (-1);
```

```
          }
          show_anal_points();
          fsum = fopen("dsum.dat","w");
          calc_radius();
          plotimage();
          fclose(fsum);
          mode= _SRES256COLOR;
          _setvideomode( mode );
          gray_scale();
          main_menu();
break;
        case 'l':
        case 'L':
         /* L : Line Analysis */
         clear_bottom();
         make_line_grid();
         main_menu();
        break;
        case 'w':
        case 'W':
_settextcolor(255);
_settextposition(35,70);
_outtext("COMPUTING ROUND GRID");
          make_round_grid();
_settextposition(35,70);
_outtext("                    ");
         main_menu();
        break;
    case 'a':
    case 'A':

         analyze();
_settextcolor(255);
         resp = getch();
         if ( (resp == 'q') || (resp == 'Q') )
            done = TRUE;
         else {
            mode= _SRES256COLOR;
            _setvideomode( mode );
            _getvideoconfig( &vc);
            gray_scale();
            plotinit2(initnum);
            plot_er();
            _setwritemode(_GXOR);
                main_menu();
         }

    break;
    case 'c':
    case 'C' :

         mode= _SRES256COLOR;
         _setvideomode( mode );
         _getvideoconfig( &vc);
```

```
filenum = 0;
saveit = 0;
gray_scale();
plotinit2(initnum);

ox = x = xlim/2;
oy = y = ylim/2;
dx = 1;
dy = 1;
prev_bn = 0;
prev_bx = 0;
prev_by = 0;
num_anal_pts = 0;

_setwritemode(_GXOR);
_setcolor( 255 );
_moveto(xxmin,oy);
_lineto(xlim,oy);
_moveto(ox,yymin);
_lineto(ox,ylim);

break;
case 'e' :
case 'E' :
if ( (num_anal_pts >= 1) && (grid_num ==RECT_REGIONS) ) {
_settextcolor(255);
_settextposition(35,70);
_outtext("DELETING LAST ANALYSIS POINT");
num_anal_pts--;
xl = anal_ptsx[num_anal_pts] - anal_sizex[num_anal_pts];
xr = anal_ptsx[num_anal_pts] + anal_sizex[num_anal_pts];
yt = (anal_ptsy[num_anal_pts] - anal_sizey[num_anal_pts]) * 2;
yb = (anal_ptsy[num_anal_pts] + anal_sizey[num_anal_pts]) * 2;
_setwritemode(_GXOR);
_setcolor(255);
_moveto(xl,yb);
_lineto(xl,yt);
_lineto(xr,yt);
_setcolor(0);
_lineto(xr,yb);
_lineto(xl,yb);
_settextposition(35,70);
_outtext("                    ");
_setcolor( 255 );
main_menu();
}
break;

} /* SWITCH (KEY) */
} /* If Key */


bn = CheckPosition(&bx, &by);
```

```
    /* CENTER BUTTON -> MARK WAFER CENTER */
    if (bn == CENTER) {
            find_center(centered);
            main_menu();
    }

    /* Right Button -> Make an Area Analysis Grid */
    if (bn == RIGHT)  {
            clear_bottom();
            make_area_grid();
_settextcolor(255);
_settextposition(34,0);
_outtext("Rectangle Draw Mode\n");
    }

    /* In the middle ? */
    if ( (bn == LEFT) && (prev_bn == LEFT) ) {
      if ( (prev_bx != bx) && (prev_by != by) ) {
  _setwritemode(_GXOR);
       _rectangle(_GBORDER, (short) prev_bx, (short) prev_by,
       (short) x, (short) y);
       _rectangle(_GBORDER, (short) prev_bx, (short) prev_by,
       (short) x, (short) y);
      }
    }

    /* Wait until a button is pressed and mark its location */
    if ( (bn == LEFT) && (prev_bn == 0) ) {
       prev_bn = bn;
       prev_bx = x;
       prev_by = y;
    }

    /* Was the button just released ??? */
    if ( (bn == 0) && (prev_bn == LEFT) ) {
      if (rotate == FALSE) {
              _setwritemode(_GPSET);
              _rectangle(_GBORDER, (short) prev_bx, (short) prev_by,
              (short) x, (short) y);
     _setwritemode(_GXOR);
              anal_type[num_anal_pts] = 0;
      } else {
        x0 = prev_bx - prev_bx;
        yy0 = prev_by - prev_by;
        x0 = (int) ( x0 * cos(theta2) - yy0 * sin(theta2));
        yy0 = (int) ( x0 * sin(theta2) + yy0 * cos(theta2));
        x0 = x0 + prev_bx;
        yy0 = yy0 + prev_by;

        x2 = x - prev_bx;
        y2 = prev_by - prev_by;
        x2 = (int)( x2 * cos(theta2) - y2 * sin(theta2));
        y2 = (int)( x2 * sin(theta2) + y2 * cos(theta2));
        x2 = x2 + prev_bx;
```

```
            y2 = y2 + prev_by;

            x3 = x - prev_bx;
            y3 = y - prev_by;
            x3 = (int)( [illegible] * cos(theta2) - y3 * sin(theta2));
            y3 = (int)( x3 * sin(theta2) + y3 * cos(theta2));
            x3 = x3 + prev_bx;
            y3 = y3 + prev_by;

            x4 = prev_bx - prev_bx;
            y4 = y - prev_by;
            x4 = (int)( x4 * cos(theta2) - y4 * sin(theta2));
            y4 = (int)( x4 * sin(theta2) + y4 * cos(theta2));
            x4 = x4 + prev_bx;
            y4 = y4 + prev_by;

         _setwritemode(_GXOR);
            _moveto((short)x0, (short)yy0);
            _lineto((short)x2, (short)y2);
            _lineto((short)x3, (short)y3);
            _lineto((short)x4, (short)y4);
            _lineto((short)x0, (short)yy0);

            anal_type[num_anal_pts] = 1;
          }
//        Save The Data For use in ANALPTS.DAT
          anal_ptsx[num_anal_pts] = (prev_bx + x) / 2;
          anal_ptsy[num_anal_pts] = (prev_by + y) / 4;
          anal_sizex[num_anal_pts]= abs(prev_bx - x)/2;
          anal_sizey[num_anal_pts]= abs(prev_by - y)/4;
          num_anal_pts++;
          _setcolor(0);
        _settextposition(35,70);
        _outtext("                        ");
        _setcolor( 255 );
          main_menu();
          prev_bn = 0;
          prev_bx = 0;
          prev_by = 0;
        }
    } while (!done);

    _moveto(xxmin,y);
    _lineto(xlim,y);
    _moveto(x,yymin);
    _lineto(x,ylim);

   _setvideomode( _DEFAULTMODE);
   exit(0);

 } /* IF RESP */

}
```

```
      program ccdanal
*
*     CCDAnal2.C700.f
*
*     control ccd post etch data analysis
*
*     tjd 4/13/92
*
      integer  nmax, ptmax, ifmax, nnmax

*     Number of Analysis Points
      parameter(nmax=724)

*     Number of Data (Time) points
*     parameter(ptmax=171)
*     parameter(ptmax=150)
      parameter(ptmax=171)

*     Number for FFT
*     parameter(ifmax=16400)
*     parameter(nnmax=13)
      parameter(ifmax=8200)
      parameter(nnmax=12)
*
      integer  numpts, i, j, cnt, start, nfit, numext
      integer pow2, nn, actpts, k, fmaxi(nmax), nlim
*
*     numpts  = Number of analysis points on the wafer considered
*     ptmax   = maximum number of analysis points on the wafer allowed
*     nmax    = maximum number of images allowed
*     i,j     = Loop counters
*     cnd     = Number of images taken
*     start   = 1st image with non-zero data in it
*     nfit    = number of points to use in derivative fitting routine
*     numext  = number of extrema for intgerferometry
*     nn      = Power of 2 for FFT
*

      real*4 r(nmax), th(nmax), y(ptmax, nmax), t(ptmax)
      real*4 yp(ptmax, nmax)
      real*4 ym(ptmax), b1m(ptmax), b2m(ptmax), ypm(ptmax)
      real*4 b, a, aa, bb, dum, siga, sigb, chi2, q, pts
      real*4 ycall(ptmax), m1, m2, int1, int2
      real*4 diff, olddiff, prevmax, prevmin
*     real*4 exttim(50), exttyp(50), delta(50), acexttim(50)
      real*4 xx(ptmax), yy(ptmax), xcent, ycent, max
      real*4 fft(ifmax), ff, fpow2, mag, freq(ifmax)
      real*4 fmax(nmax), fmag(ifmax,nmax), sum, sdelta
      real*4 xpix, ypix, theta, xxx, yyy, tlim
*
*     pts     = number of points to use in FFT smooth routine
*     r       = array of radii of analysis locations
*     th      = array of angles of analysis locations
*     y       = array of data values (image, analysis point)
```

```
*	yp	= array of smoothed data values
*	t	= array of time values
*	ym	= Max of y data.
*	ypm	= Max of smoothed y data.
*	b1m,b2m		= Max of Slopes
*	p1,p2	= print out values p2 = 1st der, p1 = 2nd der.
*	a1,b1	= Intercept & Slope for linear fit
*	a2	= Intercept for linear fit
*	bcall,
*	ycall	= 1 Dimensional Array for routine calls.
*	diff,
*	olddiff = vars for peak locator.
*	exttim  = extrema times for interferometry
*	acextim = Actual Extrema times.
*	exttyp  = extrema types for interferometry (+1 = max, -1 = min)
*	prevmax = time point of previous maximum for interferometry.
*	prevmin = time point of previous minumum for interferometry.
*	delta	= Time interval for interferometry.
*	fft	= Temp. array for fft processing
*	ff	= calculated frequency
*	fpow2	= floating value of 2**nn for fft
*	mag	= magnitude of FFT
*

	character*80 fname, fname2
*
	write(6,*) 'CCD DATA ANALYSIS PACKAGE'
	write(6,*) 'T.J.Dalton MIT 66-225 258-8840'
	write(6,*)
	write(6,*) 'Maximum number of data points     = ',nmax
	write(6,*) 'Maximum number of analysis points = ',ptmax
	write(6,*)
	write(6,*)
*
	write(6,*)
	write(6,*) 'Use Default Names ? (Y/N) : '
	read(5,13) fname
*
	if (fname .eq. 'n') then

3	  continue

	    write(6,*)
	    write(6,*) 'Enter the DSUM.DAT file name: '
	    read(5,13) fname
13	    format(a80)
	    open(unit=1, file=fname, status='old', err = 12)
	    goto 14
12	    continue
	    write(6,*) 'FILE NOT FOUND'
	    goto 3
14	    continue
```

```
4           continue

            write(6,*)
            write(6,*) 'Enter the ANALPS.DAT file name: '
            read(5,13) fname
            open(unit=8, file=fname, status='old', err = 22)
            goto 24
22          continue
            write(6,*) 'FILE NOT FOUND'
            goto 4
24          continue

        else
          write(6,*) 'Opening Default Files, in /c700/src/video'
          open(unit=1, file='/metl/metl/c700/src/video/dsum.dat', status='old')
          open(unit=8, file='/metl/metl/c700/src/video/analpts.dat', status='old')
        endif
        read(8,*) numpts
*
        if (numpts .gt. nmax) then
                write(6,*)
                write(6,*) 'Number of Analysis Points > NMAX'
                write(6,*)
                write(6,*) numpts,nmax
                write(6,*)
                write(6,*) 'JOB CANCELLED'
                write(6,*)
                stop
        endif

        do 77, i=1, numpts
77      read(8,*) xx(i), yy(i), dum, dum, dum
        read(8,*) xcent, yxent
        read(8,*) xpix, ypix, theta
        close(unit=8)

*
*
*
*       Determine the Number of Wafer Points Analyzed
*
***     write(6,*) 'Enter Number of Analysis Points'
***     read(5,*) numpts
        write(6,*) 'Number of Analysis Points = ',numpts

*
*
*       Read in the Radii and Angles of the points
*
        read(1,*) (r(i),i=1,numpts)
        read(1,*) (th(i),i=1,numpts)
*
        write(6,*) 'Radii and Angles read in'
*
```

```
*
*       Now, Read in the points
*
        start = 0
        cnt = 1

100     continue
        read(1,*,err=110) t(cnt), (y(cnt,j),j=1,numpts)
        if ( (start .eq. 0) .and. (y(cnt,1) .ne. 0) ) start = cnt
*       if (t(cnt) .gt. 60.0) goto 110
        cnt = cnt + 1
        if (cnt .gt. ptmax) then
                write(6,*)
                write(6,*) 'Number of Points Read in > PTMAX'
                write(6,*)
                write(6,*) 'JOB CANCELLED'
                write(6,*)
                stop
        endif
        goto 100

110     cnt = cnt - 1
        write(6,*) 'Finished reading in ',cnt,' data points'
        close(unit=1)
*
*
        write(6,*)
        write(6,*) 'Time Start = ',t(start)
        write(6,*)
        write(6,*) 'Enter the Time to Start at: '
        read(5,*) tlim
        write(6,*) 'Enter the Time to Start at is ',tlim

        i = start-1
219     continue
          i = i + 1
          if (t(i) .lt. tlim) then
            if (i .lt. cnt) then
              goto 219
            else
                  start = i
              goto 221
            endif
          else
            start = i
            goto 221
          endif

221     continue
         write(6,*)
         write(6,*) 'Tstart Found at Point ',start,' which is t = ',t(start)
         write(6,*)
*
         write(6,*)
```

```
      write(6,*) 'Time Limit = ',t(cnt)
      write(6,*)
      write(6,*) 'Entεr the Time to Cut Off at: '
      read(5,*) tlim
      write(6,*) 'Enter the Time to Cut Off at is ',tlim

      i = start-1
119   continue
        i = i + 1
        if (t(i) .lt. tlim) then
          if (i .lt. cnt) then
            goto 119
          else
                cnt = i
            goto 121
          endif
        else
          cnt = i
          goto 121
        endif

121   continue
      write(6,*)
      write(6,*) 'Tlim Found at Point ',cnt,' which is t = ',t(cnt)
      write(6,*)
*
      nn = 0
      sum = 0.0
      do 111, i=2, cnt
        nn = nn + 1
        sum = sum + (t(i) - t(i-1))
111   continue
*
      sdelta = sum/floati(nn)
*
      write(6,*)
      write(6,*) 'Sampling Rate = ',1./sdelta,' Hz','   Sampling Time = ',sdelta,' sec'
      write(6,*)

*
*     write(6,*) 'Enter # Pts to Fit Straight Line
*    1to Above & Below pt i ??? '
*     read(5,*) nfit
      nfit = 3
      write(6,*) '# Pts to fit = ',nfit
*
*     write(6,*) 'Enter # of Pts to Smooth ? '
*     read(5,*) pts
      pts = 5
      write(6,*) '# Smooths = ',pts
*
*
*
*     Calculate stuff for the FFT
```

```
*
      i=1
      actpts = cnt-start+1
300   continue
       i = i + 1
       pow2 = 2**i
*
*      this must be integer division! (DIV)
*
       if ( (actpts/pow2) .gt. 1) goto 300
         if ( ( (actpts/pow2) .eq. 1) .and.
  1             ( mod(actpts,pow2) .ne. 0) ) goto 300
*
*      go up an extra power, pad it out with zero's
*
       nn = i + 3
       if (nn .gt. nnmax) nn = nnmax
       write(6,*) 'FFT Size desired is 2^',nn
**********{}{}{}{}{}{}{}{}{}{}
**********{}{}{}{}{}{}{}{}{}{}
       write(6,*) 'ENTER NN to use'
       read(5,*) nn
**********{}{}{}{}{}{}{}{}{}{}
**********{}{}{}{}{}{}{}{}{}{}
       if (nn .gt. nnmax) nn = nnmax
*
       pow2 = 2**nn
       fpow2 = floati( pow2 )
       j = 1
      open(unit=13,file='fout.dat')
       do 311, i=pow2+1, 2*pow2, 2
         freq(j) = floati( ( i - (2*pow2+1) )/2 ) /fpow2
      write(13,*) j,freq(j)
311      j = j + 1
       do 312, i=1, pow2+1, 2
         freq(j) = floati( (i-1)/2 ) / fpow2
      write(13,*) j,freq(j)
312      j = j + 1

*
       write(6,310) actpts,nn,pow2
310    format(' Actual Nonzero Points = ',i3,'  FFT-n = ',i2,'  2**N = ',i4)
*
*
*
*     take derivatives of laser interferometry
*     signals to see the results...
*
*     Based upon an idea in JVSTB 9(5), 2497.
*
*
*
      open(unit=2, file='ccdanal.dat')
      open(unit=8, file='dsurf.dat')
```

```
        open(unit=9, file='ccdfft.dat')
*       write(9,342)
*342    format(' Frequency      Real       Imaginary      Magnitude')
*         x1234567890xxx12345678901234xxx12345678901234xxx12345678901234
***     do 313, i=1, pow2+1
***313  write(9,314) freq(i)
***314  format(f10.8)
*
*
*
*       Do a Baseline Subtraction
*
        write(6,*) 'Doing Baseline Subtraction & Smoothing to yp'

        do 5, j=1, numpts

          if ((mod(J,100)) .eq. 0) write(6,*) 'Starting J = ',j
*
*         transfer y data to a 1-d array
          do 6, i=start, cnt
6         ycall(i) = y(i,j)

          call fit( t(start), ycall(start), (cnt-start), dum, 0,
   1      a1, b1, siga, sigb, chi2, q)
*
          ym(j) = -999.9
          do 10, i=start, cnt, 1
            yp(i,j)  = y(i,j)
            y(i,j)   = y(i,j) - (a1 + b1 * t(i))
            if (y(i,j) .gt. ym(j)) ym(j) = y(i,j)
10        continue
*
*
*         Take the FFT of the data
          do 319, k=1, 2*pow2+1
319       fft(k) = 0.0
*
*         Transfer Data to FFT Array
*
          k = 1
          do 320, i=start, cnt, 1
            fft(k)   = y(i,j)
            fft(k+1) = 0.0
            k = k + 2
320       continue
*
*         if the data isn't in powers of 2, fill out to the next power of 2 with 0's
*
          do 330, i=actpts+1, pow2, 1
            fft(k)   = 0.0
            fft(k+1) = 0.0
            k = k + 2
330       continue
*
```

```
      k = 1
      do 331, i=start, start+5
        fft(k)   = 0.0
        fft(k+1) = 0.0
        k = k + 2
331   continue
*
      call four1(fft, pow2, 1)
*
*     print out the spectrum from -1/2 -> -1/N
*
      k = 1
*
*     check 1/2 the spectrum for the dominant spectral component
*
      fmaxi(j) = 0
      max = 0.0
      do 360, i=pow2+1, 2*pow2, 2
        fmag(k,j) = sqrt( fft(i)*fft(i) + fft(i+1)*fft(i+1))
        if (fmag(k,j) .gt. max) then
              max = fmag(k,j)
              fmaxi(j) = k
        endif
        k = k + 1
360   continue

*
*     now that we have a candidate for the maximum, lets find it!
*
      ycall(1) = fmag( fmaxi(j)-2, j)
      ycall(2) = fmag( fmaxi(j)-1, j)
      ycall(3) = fmag( fmaxi(j)  , j)
*
      call fit( freq( fmaxi(j)-2), ycall(1), 3, dum, 0,
     1 a, b, siga, sigb, chi2, q)
*
      ycall(1) = fmag( fmaxi(j)  , j)
      ycall(2) = fmag( fmaxi(j)+1, j)
      ycall(3) = fmag( fmaxi(j)+2, j)
*
      call fit( freq( fmaxi(j)), ycall(1), 3, dum, 0,
     1 aa, bb, siga, sigb, chi2, q)
*
      fmax(j) = abs( (aa-a)/(b-bb) )
*
*     print the spectrum from 0 to 1/2
*
      do 370, i=1, pow2+1, 2
        fmag(k,j) = sqrt( fft(i)*fft(i) + fft(i+1)*fft(i+1))
        k = k + 1
370   continue
*
*
*
```

```
5        continue


*
*        print out the results of the calculations
*
*
         if (numpts .gt. 5) then
            nlim = 9
         else
            nlim = numpts
         endif
*
         do 85, i=start, cnt, 1
*
           write(2,95) t(i), ( (yp(i,j)),j=1,nlim)
*          write(2,95) t(i), ( (y(i,j)/ym(j)),j=1,nlim)
95       format(f6.2,20(2x,f9.4))
85       continue
*
      do 313, i=1, pow2+1
313     write(9,314) freq(i), (fmag(i,j),j=1,nlim)
314     format(f10.8,20(3x,f14.7))
*
*
         pi = 4.0 * atan(1.0)
         do 200, j=1, numpts,1
                 xxx = r(j) * cos( th(j) * pi / 180.0 + theta)
                 yyy = r(j) * sin( th(j) * pi / 180.0 + theta)
                 er = 7534.0/2.00/3.42*60.0 / sdelta * fmax(j)
           write(8,202) r(j), th(j), xxx, yyy, 1./fmax(j), j, er
202        format(f6.3,2x,f7.2,2x,2(f6.3,2x),f10.5,5x,i3,5x,f7.0)
200      continue
*
*
         close(unit=2)
         close(unit=8)
         close(unit=9)
*
         end
*
         include "lifanal.f"
         include "smooft.f"
         include "realft.f"
         include "four1.f"
         include "gser.f"
         include "gcf.f"
         include "gammln.f"
         include "gammq.f"
         include "fit.f"
*
```

Appendix 4

Interferometry Model

This appendix contains a listing of the source code used for modeling of reflectivity in interferometry. The code was written in Unix Fortran and run on a Mips M120/5.

input4.dat for Interferometry Model

```
3
1.0, 0.0
3.730, -0.018
1.45, 0.0
3.730, -0.009
5000.0
100.0
2500
618.0
135.0
1.0
7534.0
```

Code for Interferometry Model

```
        program model4
*
*********************************************************************************
*       Model Response of Observed CCD Interferometry Signal
*       to Different Film Structures
*********************************************************************************
*
*       Based upon the Response Equations Given in:
*
*       Hugo Anders, Thin Films in Optics(The Focal Press, NY, 1967)
*       pp.25-26, 47-48
*
*********************************************************************************
*       Normal Incidence Only
*       The Reflection of Light at Four Interfaces (i.e. by Three Layers)
*********************************************************************************
*
*       (c) 1994  Timothy J. Dalton
*
*       MIT, 66-225 Cambridge MA 02139
*
*       v1.0 21 Jan, 1994
*
*               Initial Coding
*
*********************************************************************************
*
*       Declare Variables
```

```
*
        integer     NLayerMax, j, NLayer
        PARAMETER  (NLayerMax = 4)

        complex*16 Refl, PI, top, bot, i, zero, rr
        complex*16 DELTA1, DELTA2, DELTA3, r(NLayerMax)
        real*8    D(NLayerMax-1), N(0:NLayerMax), K(0:NLayerMax)
        real*8    ER(NLayerMax-1), dD(NLayerMax-1)
        real*8      LAMBDA, Tlim, Tstep, Time
        complex*16 Nc(0:NLayerMax)
*
*       Declare the constant, i, where i*i = -1
*
        i   = (0.0d0, 1.0d0)
        zero = (0.0d0, 0.0d0)
*
*       Main Program
*
*       Compute Pi
*
        PI = dcmplx(4.0*datan(1.0d0),0.0)
*
        write(6,1000)
1000    format(//'CCD Interferometry Signal Modeling'//)
*
*       Open Output File
*
        open(unit=1,file='signal.dat')
*
*       Index of Refrction of films.
*       Index = (N + iK)
*
*       0 - Air
*       1 - Polysilicon
*       2 - Silicon Dioxide
*       3 - Silicon Wafer
*
        open(unit=2,file='input4.dat',status='old')
*
*       Determine the Number of Layers in the Film
*
        read(2,*) Nlayer
        if (NLayer .gt. 4) then
                write(6,4) NLayer, NLayerMax
4               format(///'ILLEGAL VALUE OF NLAYER.  NLAYER= ',i2,
   1            ' and Must be <= ',i2///)
                stop
        endif
*
        do 5, j=0, NLayer
          read(2,*) N(j),K(j)
          Nc(j) = dcmplx(N(j),K(j))
          write(6,*) 'Nc(j) = ',j,Nc(j)
5       continue
```

```
*
*       Film Thicknesses in Angstroms.
*
        do 6, j=1, NLayer-1
          read(2,*) D(j)
          write(6,*) 'D(j) = ',j,D(j)
6       continue
*
*       zero Unused Layers
*
        do 12, j=NLayer, NLayerMax-1
          D(j) = 0.0
          write(6,*) 'D(j) = ',j,D(j)
12      continue
*
*       Etching Rates  - Angstroms/Second
*
        do 7, j=1, NLayer-1
          read(2,*) ER(j)
          ER(j) = ER(j) / 60.0
          write(6,*) 'ER(j) = ',j,ER(J)
7       continue
*
*       Timelimit and Timestep in Seconds.
*
        read(2,*) Tlim
        write(6,*) 'Tlim = ',Tlim
        read(2,*) Tstep
        write(6,*) 'TStep = ',TStep
*
*       Wavelength of Analysis Light in Angstrongs
*
        read(2,*) LAMBDA
        write(6,*) 'Lambda = ',LAMBDA
*
        close(unit=2)
*
*       Change in Film Thickness during 1 Timestep
*
        do 8, j=1, NLayer-1
          dD(j) = ER(j) * TStep
8       continue
*
*       Calculate Some Values that do not change with film thickness.
*
        do 9, j=1, NLayer
          r(j) = (Nc(j-1) - Nc(j)) / (Nc(j-1) + Nc(j))
          write(6,*) 'r(j) = ',j,r(j)
9       continue
*
*       Zero Out the Nonused Layers!
*
        do 11, j=NLayer+1, NLayerMax
          r(j) = Zero
```

```
          write(6,*) 'r(j) = ',j,r(j)
11      continue
*
        r1 = r(1)
        r2 = r(2)
        r3 = r(3)
        r4 = r(4)
*
*       Now, Follow the signal as a function of time
*
        DO 20, Time=0.0, Tlim, Tstep
*
*         This Equation is Valid for Nlayer = NLayerMax
*         if NLayer < NLayerMax, this reduces to the proper
*         solution by setting r(k) = 0 where k=Nlayer+1..NLayerMax
*
          DELTA1 = -i * (4.0*PI/LAMBDA * Nc(1) * D(1))
          DELTA2 = -i * (4.0*PI/LAMBDA * Nc(2) * D(2))
          DELTA3 = -i * (4.0*PI/LAMBDA * Nc(3) * D(3))
        if (time .eq. 0.0) then
                write(6,*) 'Delta1 = ',delta1
                write(6,*) 'Delta2 = ',delta2
                write(6,*) 'Delta3 = ',delta3
        endif
*
************************************************************************
          top = r1 + r2*zexp(DELTA1) + r3*zexp(DELTA1+DELTA2) +
   1         r4*zexp(DELTA1+DELTA2+DELTA3) + r1*r2*r3*zexp(DELTA2) +
   2             r1*r3*r4*zexp(DELTA3) + r1*r2*r4*zexp(DELTA2+DELTA3) +
   3             r2*r3*r4*zexp(DELTA1+DELTA3)
*
************************************************************************
          bot = 1 + r1*r2*zexp(DELTA1) + r1*r3*zexp(DELTA1+DELTA2) +
   1             r1*r4*zexp(DELTA1+DELTA2+DELTA3)+ r2*r3*zexp(DELTA2) +
   2             r3*r4*zexp(DELTA3) + r2*r4*zexp(DELTA2+DELTA3) +
   3             r1*r2*r3*r4*zexp(DELTA1+DELTA2)
************************************************************************
*
*         Now, this is the resultant wave...
*
          rr = top / bot
*
*         But, we are interested in the measureable Quantity....Reflectance
*
*         So, Calculate the Observed Reflectance
*
          Refl = rr * dconjg(rr)
*
          write(1,10) time, Refl, D(1), D(2), D(3)
10        format(f8.2,2x,2(f20.12,2x),3(2x,f7.1))
*
          if (D(1) .gt. 0.0) then
                D(1) = D(1) - dD(1)
          else if (D(2) .gt. 0.0) then
```

```
            D(2) = D(2) - dD(2)
      else if (D(3) .gt. 0.0) then
            D(3) = D(3) - dD(3)
      else
            D(1) = 0.0D0
            D(2) = 0.0D0
            D(3) = 0.0D0
      endif

20    CONTINUE

30    CONTINUE
      close(unit=1)

      END
```

Appendix 5

CSTR Model Code

This appendix contains a listing of the source code used for a cstr model in Chapter 8. The Code was written in Unix Fortran and run on a Mips M120/5.

Code for CSTR Model

```
      program cstr7
*
*     Model Gas Phase Kinetics in AME-5000 for very simple
*     system.
*
      integer ntrial, n, np, nrun, jjj, j, i
      double precision tolx, tolf, temp

      parameter (ntrial=100, tolx=1.0e-6, n=5, tolf=1.0e-6, np=15)
      parameter (nrun = 5)

      double precision x(np), alpha(np,np), beta(np)
*
      common /props/ totflow, cl2flow, er, t, p, nsi, dw, pi, a, v,
   1 kb, mwcl, ne, gamsicl2, kf, kd, Nin, Csicl2, Ketch, Mwsi, Nav,
   2 Ccl2, gamcl2, Ccl, gamcl, Tau, Qf

      double precision totflow(np), cl2flow(np), er(np), T, P
      double precision Nsi, dw, pi, A, V, kb, mwcl, ne, gamsicl2, kf, kd
      double precision Nin, Csicl2, ketch, Mwsi, Nav, Ccl2, gamcl2
    double precision Ccl, gamcl, Tau, Qf
      double precision prob, cl
*
      write(6,*) 'AME5000'
*
      totflow(1) = 13.0
      totflow(2) = 20.0
      totflow(3) = 40.0
      totflow(4) = 80.0
      totflow(5) = 116.0
*
      cl2flow(1) = 10.0
      cl2flow(2) = 15.0
      cl2flow(3) = 30.0
      cl2flow(4) = 60.0
      cl2flow(5) = 87.0
*
      er(1) = 4322.0
      er(2) = 4378.0
      er(3) = 5295.0
      er(4) = 5900.0
      er(5) = 6311.0
*
      T = 500.0
```

```
      P = 0.1
*
      Nsi = 5.00e22
      dw  = 10.0
*
      pi = 4.0 * atan(1.0)
      A  = pi/4.0*dw*dw
      V = 13000.0
*
      kb = 1.380e-16
      mwcl = 35.453
      mwsi = 28.0855
      Nav = 6.022e23
*
      Csicl2 = sqrt( 8.*kb*T/(pi*(mwsi+2.*mwcl)/Nav))
      Ccl2 = sqrt( 8.*kb*T/(pi*(2.*mwcl)/Nav))
      Ccl = sqrt( 8.*kb*T/(pi*(mwcl)/Nav))
*
*     Assumptions
*
      kf = 3.48e-9
      ne = 1e10
      gamsicl2 = 0.0
      gamcl2 = 0.3
      gamcl  = 1.0
      kd = 1e-10
*
      open(unit=2,file="cstr7.out")
      open(unit=3,file="cstr7.dat")
*
      write(2,*) 'CSTR Modeling'
      write(2,*) 'A = ',A
      write(2,*) 'V = ',V
      write(2,*)

      do 100, jjj=1, nrun
*
        write(2,*) 'RUN = ',jjj
        write(2,*) 'ER = ',er(jjj),' Total Flow = ',totflow(jjj)
        write(2,*)
        write(2,*) 'Csicl2 = ',Csicl2,' cm/s'
        write(2,*) 'Ccl2   = ',Ccl2,' cm/s'
        write(2,*) 'Ccl    = ',Ccl,' cm/s'
        write(2,*) 'kd = ',kd
*
*       initial guesses
*
*
*       Qf = cl2flow(jjj)*760.0/P*T/273.15/60.0
        Qf = totflow(jjj)*760.0/P*T/273.15/60.0
        Tau = V / Qf
*
        write(2,*) 'Qf  = ',Qf
        write(2,*) 'Tau = ',tau
```

```
*
*        Nin = 2.69e19/60.0 * cl2flow(jjj)
         Nin = 2.69e19/60.0 * totflow(jjj)
*
         ketch = er(jjj)*Nsi/60.e8
*
         write(2,*) 'Nin = ',nin
         write(2,*) 'ketch = ',ketch
*
         temp = P*1.0e7/(760.0*Kb*T)
*
*        Cl2, SiCl4, SiCl2, Cl, Qexit
*
         x(1) = temp / 10.0
         x(2) = temp /2.0
         x(3) = ketch*A/V*Tau
         x(4) = x(3)
         x(5) = Qf
         call usrfun(x,alpha,beta)
*
         do 11,i=1,n
11          write(2,21) i,x(i),-beta(i)
*
         do 13, j=1, ntrial
            call mnewt(10,x,n,tolx,tolf)
            call usrfun(x,alpha,beta)
                 write(2,*)
                 write(2,*) 'After Ntrial = ',j
                 do 20, i=1, n
                    write(2,21) i,x(i),-beta(i)
21                     format(i2,2x,e12.3,2x,e12.3)
20               continue
13       continue

*
*        Estimate a Reaction Probability from Flux, and
*        From ER
*
*        flux of cl to surface in Molecules/cm^2/s
*
         cl = 2. * (x(1)*Ccl2/4.) + (x(2)*Ccl2/4.)
*
*        Flux of SiCl4 Away from Surface = Ketch
*
*        Flux of Cl consumed = 4*Ketch
*
*        Rxn Prob. = Cl Consumed / Cl Arriving
*
         prob = 4.*Ketch / Cl
*
        write(6,*) cl2flow(jjj),prob
         write(3,30) cl2flow(jjj),(x(i),i=1,n),prob
30       format(f5.1,6(2x,e14.5))
*
```

```
100     continue

        end

        subroutine usrfun(x,alpha,beta)
        integer np
        parameter(np=15)
        double precision alpha(np,np), beta(np), x(np)

        common /props/ totflow, cl2flow, er, t, p, nsi, dw, pi, a, v,
    1  kb, mwcl, ne, gamsicl2, kf, kd, Nin, Csicl2, Ketch, Mwsi, Nav,
    2  Ccl2, gamcl2, Ccl, gamcl, Tau, Qf

        double precision totflow(np), cl2flow(np), er(np), T, P
        double precision Nsi, dw, pi, A, V, kb, mwcl, ne, gamsicl2, kf, kd
        double precision Nin, Csicl2, ketch, Mwsi, Nav, Ccl2, gamcl2
     double precision Ccl, gamcl, Tau, Qf

        alpha(1,1) = -x(5)/v -kf*ne -0.0*GamCl2*Ccl2*A/V/4.
        alpha(1,2) = 0.5 * Gamcl*Ccl*A/V/4.
        alpha(1,3) = kd*ne
        alpha(1,4) = 0.0
        alpha(1,5) = -x(1)/v

        alpha(2,1) =  2*kf*ne
        alpha(2,2) =  -x(5)/v -Gamcl*Ccl*A/V/4.
        alpha(2,3) =  0.0
        alpha(2,4) =  0.0
        alpha(2,5) =  -x(2)/v

        alpha(3,1) = 0.0
        alpha(3,2) = 0.0
        alpha(3,3) = -x(5)/v -kd*ne
        alpha(3,4) = 0.0
        alpha(3,5) = -x(3)/v

        alpha(4,1) = 0.0
        alpha(4,2) = 0.0
        alpha(4,3) = kd*ne
        alpha(4,4) = -x(5)/v
        alpha(4,5) = -x(4)/v

        alpha(5,1) = -1.0
        alpha(5,2) = -1.0
        alpha(5,3) = -1.0
        alpha(5,4) = -1.0
        alpha(5,5) = 0.0

*       Cl2
        beta(1) = -( Nin/V -x(5)*x(1)/v -kf*ne*x(1) +kd*ne*x(3)
    1        -Gamcl2*Ccl2*A/V/4. * x(1)
    2             +1.0*Gamcl2*Ccl2*A/V/4. * x(1) +
    3     +0.5*Gamcl *Ccl *A/V/4. * x(2)
    4     - 2.*Ketch*A/V )
```

```
*       Cl
        beta(2) = -( -x(2)*x(5)/v +2*kf*ne*x(1) -Gamcl*Ccl*A/V/4. * x(2) )

*       SiCl4
        beta(3) = -( -x(3)*x(5)/v + ketch*A/V -kd*ne*x(3) )

*       SiCl2
        beta(4) = -( -x(4)*x(5)/V + kd*ne*x(3) )

*       Qexit
        beta(5) = -(Nin/Qf - x(1) - x(2) -x(3) -x(4) )
*
        return
        end
```

Appendix 6

Other Equipment Drawings and Descriptions

A6.1 Magnetic Field Synchronization

Data acquisition was synchronized to the rotating magnetic field on the AME-5000. This served to effectively "lock" the magnetic field with respect to the data acquisition. There was spatial variations in plasma emission due to the field, but the temporal variations caused by rotation were removed from the collected data, leaving temporal variations arrising from changes within the plasma. Synchronization of data acquisition was accomplished using a Hall effect magnetic field sensor. The Hall effect is the creation of a transverse voltage difference due to charge carrier separation when a magnetic field is oriented perpendicular to a flowing current. A sensor (#SS21PE) manufactured by MICRO SWITCH of Freeport, Illinois was used. This sensor has a current sinking digital output. It is powered by 4.5 to 5.5 VDC and operates from -40° to +150°C. The sensor has a rapid response, with a 1.5 μs rise time and 0.5 μs fall time. It typically closes at 15 gauss (25 gauss maximum) and opens at 11 gauss (5 gauss minimum). The sensor can not be damaged by application of excess magnetic fields. The sensor is omnipolar; it is not sensitive to the direction of the field (north or south).

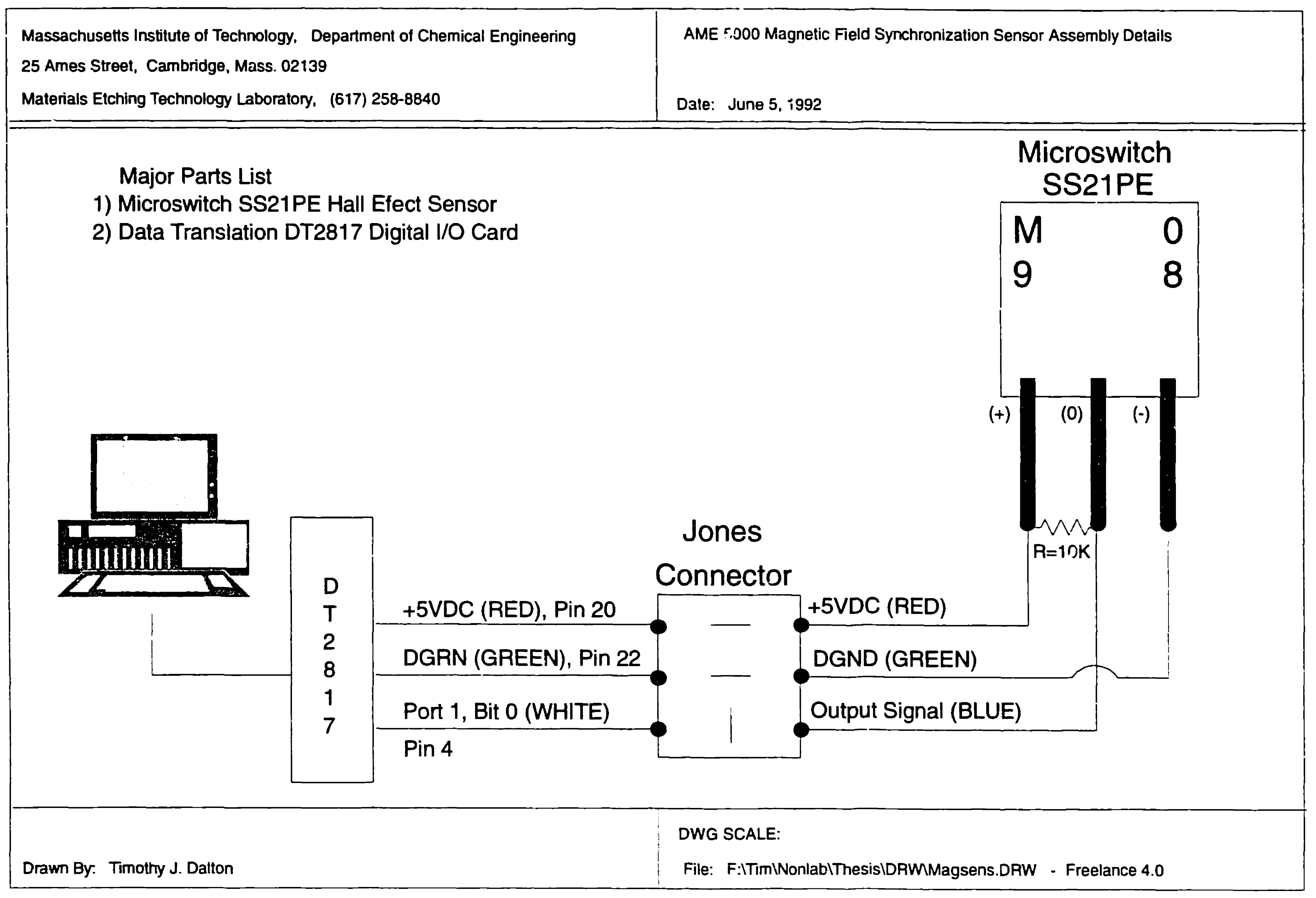
The magnetic field sensor was connected as show in the following figure. It was interfaced to the personal computer using a Data Translation DT2817 Digital Input/Output card. The DT2817 served to both provide power to the sensor and to read the output of the sensor. As the sensor is a current sink, it reads true (+5V) when no magnetic field is present, and reads false (+0.4V) when a field is present. The figure after the sensor connections shows typical output from the SS21PE. The sensor output goes high (true) as the field initially drops to zero (time t0). Data acquisition is triggered on the rise from field present to field absent at t0. As the field grows, the sensor switches low and remains so until the field begins to reverse direction. It goes high again and then low as the reversed field strength increases. Finally, as the field again decreases, the sensor again goes high at time t0+T, where T is the magnetic field rotation period, which is set by the user of the AME-5000. Data acquisition is triggered on this second rise as well. This process is repeated, starting data collection on the rise of every other high pulse from the sensor.

A6.2 Other Drawings

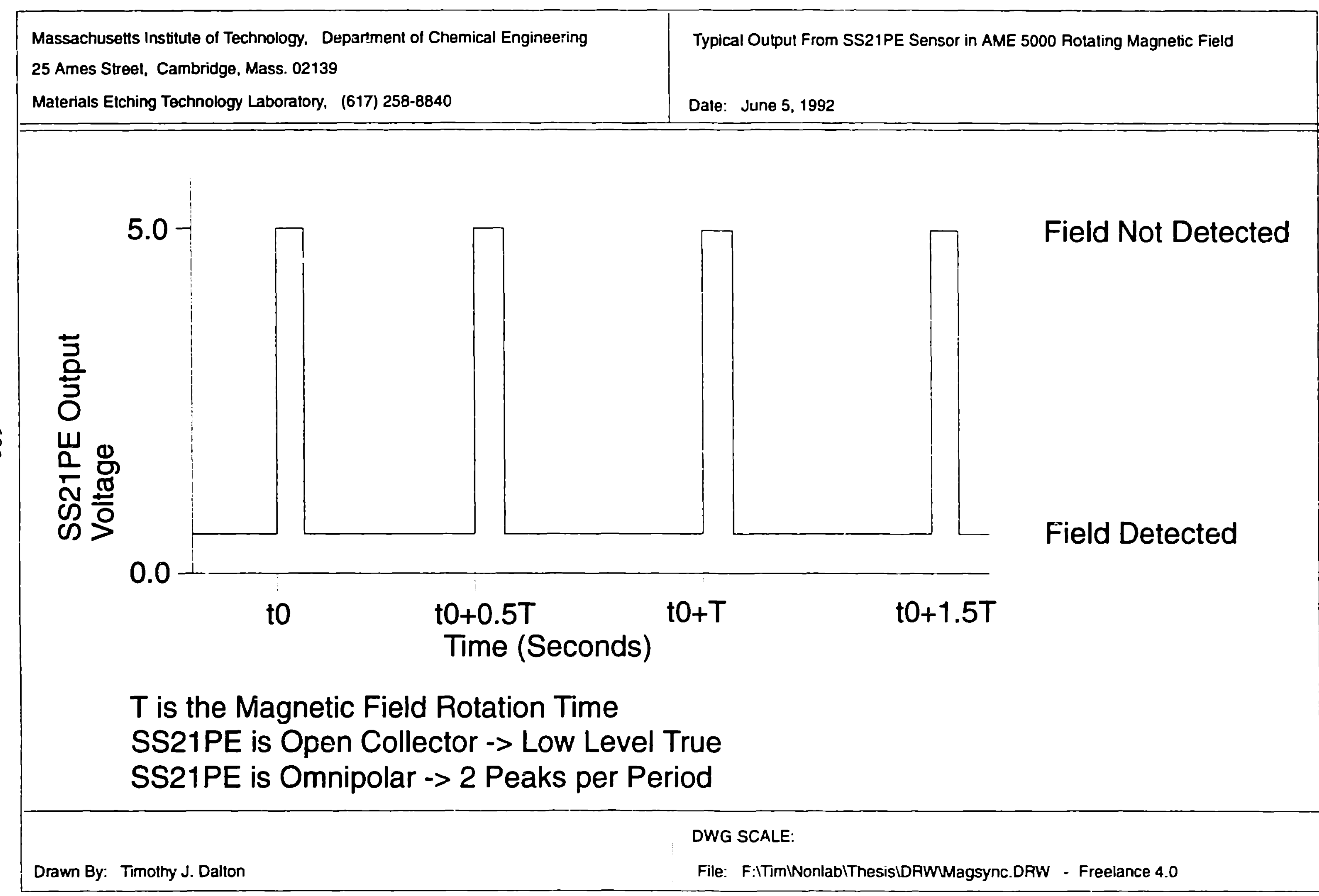
Other drawings not yet contained within the thesis are given here. They are:

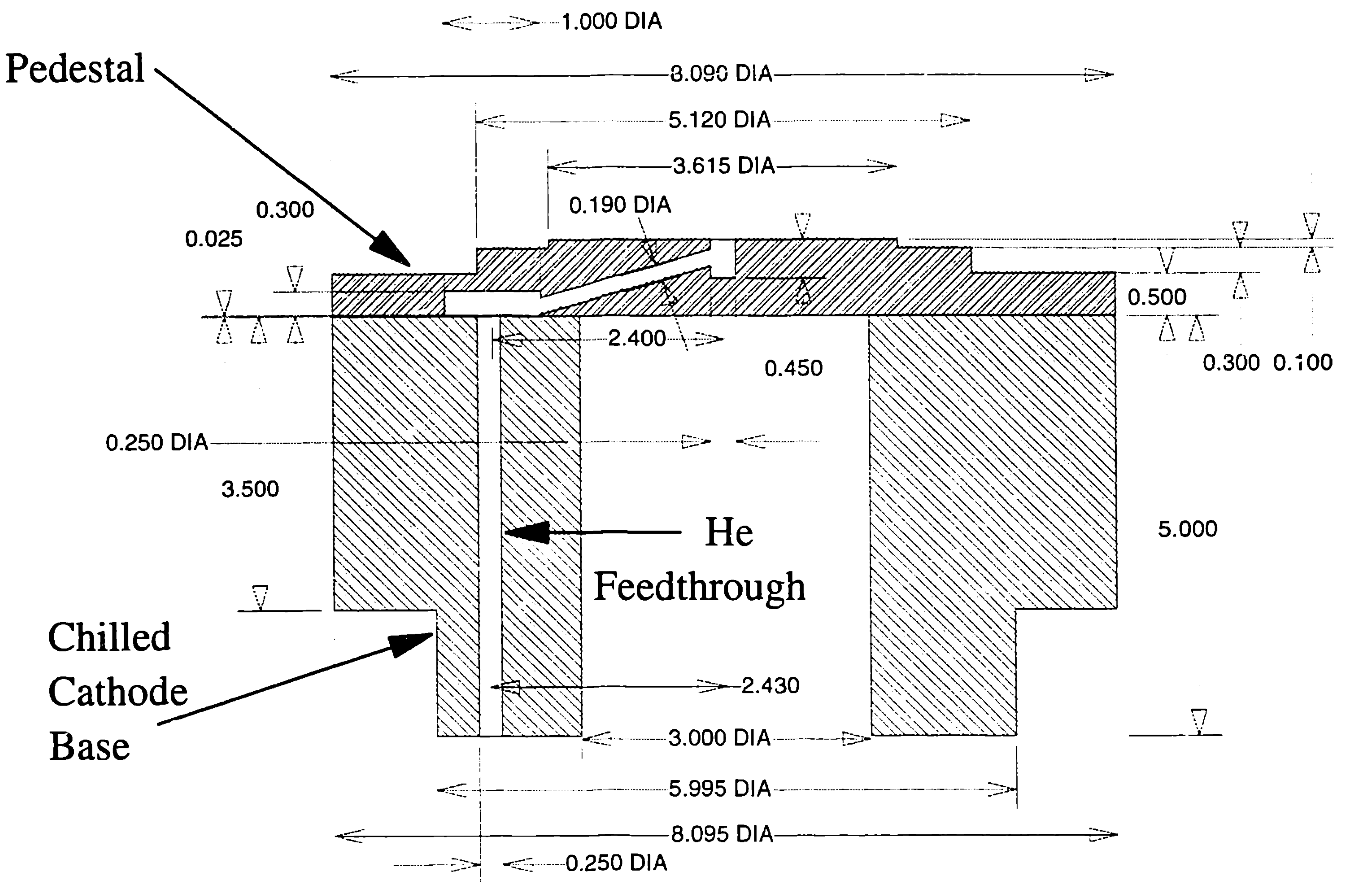
(1) AME-5000 cathode assembly with dimensions used for heat transfer analysis and for temperature probe intoduction.

(2) Feedthrough flange for the langmuir probe used in Chapter 7.

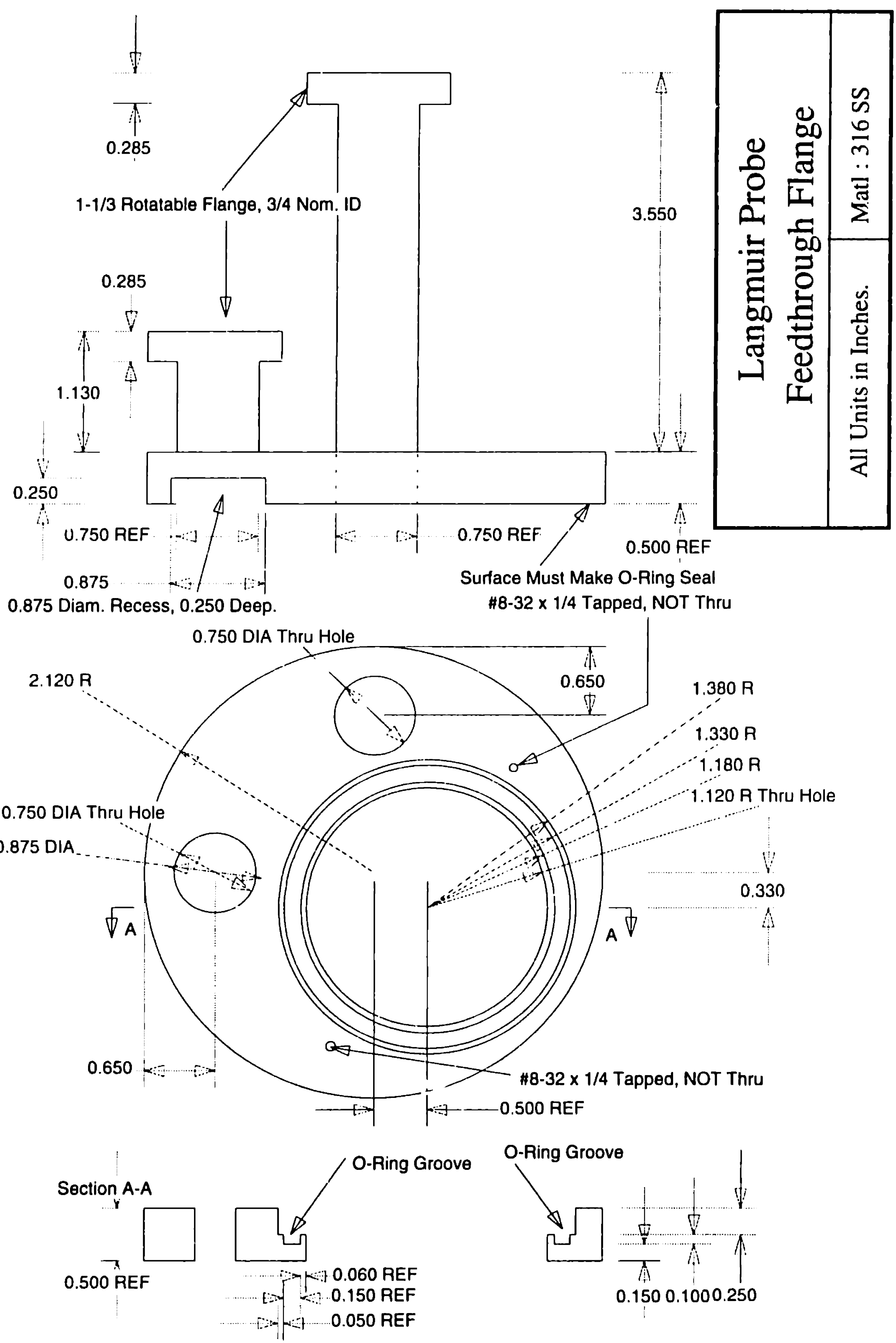


Massachusetts Institute of Technology, Department of Chemical Engineering
25 Ames Street, Cambridge, Mass. 02139
Materials Etching Technology Laboratory, (617) 258-8840
AME 5000 Magnetic Field Synchronization Sensor Assembly Details
Date: June 5, 1992
Major Parts List
1) Microswitch SS21PE Hall Efect Sensor
2) Data Translation DT2817 Digital I/O Card
Microswitch
SS21PE
M 0
9 8
(+)
(0)
(-)
R=10K
Jones
Connector
D T 2 8 1 7
+5VDC (RED), Pin 20
DGRN (GREEN), Pin 22
Port 1, Bit 0 (WHITE)
Pin 4
+5VDC (RED)
DGND (GREEN)
Output Signal (BLUE)
Drawn By: Timothy J. Dalton
DWG SCALE:
File: F:\Tim\Nonlab\Thesis\DRW\Magsens.DRW - Freelance 4.0


Massachusetts Institute of Technology, Department of Chemical Engineering
25 Ames Street, Cambridge, Mass. 02139
Materials Etching Technology Laboratory, (617) 258-8840
Typical Output From SS21PE Sensor in AME 5000 Rotating Magnetic Field
Date: June 5, 1992
SS21PE Output Voltage
5.0
0.0
Field Not Detected
Field Detected
t0
t0+0.5T
t0+T
t0+1.5T
Time (Seconds)
T is the Magnetic Field Rotation Time
SS21PE is Open Collector -> Low Level True
SS21PE is Omnipolar -> 2 Peaks per Period
DWG SCALE:
Drawn By: Timothy J. Dalton
File: F:\Tim\Nonlab\Thesis\DRW\Magsync.DRW - Freelance 4.0



AME-5000 100-mm Cathode Assembly


Langmuir Probe
Feedthrough Flange
Matl : 316 SS
All Units in Inches.
0.285
1-1/3 Rotatable Flange, 3/4 Nom. ID
3.550
0.285
1.130
0.250
0.750 REF
0.750 REF
0.500 REF
0.875
0.875 Diam. Recess, 0.250 Deep.
Surface Must Make O-Ring Seal
#8-32 x 1/4 Tapped, NOT Thru
0.750 DIA Thru Hole
2.120 R
0.650
1.380 R
1.330 R
1.180 R
1.120 R Thru Hole
0.750 DIA Thru Hole
0.875 DIA
0.330
A
A
0.650
#8-32 x 1/4 Tapped, NOT Thru
0.500 REF
Section A-A
O-Ring Groove
O-Ring Groove
0.500 REF
0.060 REF
0.150 REF
0.050 REF
0.150
0.100
0.250

A6.3 Langmuir Probe Construction

The prime component of the Langmuir probes used for the work of Chapter 8 is the inductors. These are manufactured by J.W. Miller and are obtained from:

Bell Industries (Our Customer Number 0000900)
PO Box 2859, Gardena CA, 90247-1059
(310) 515 1720

The inductors obtained were:

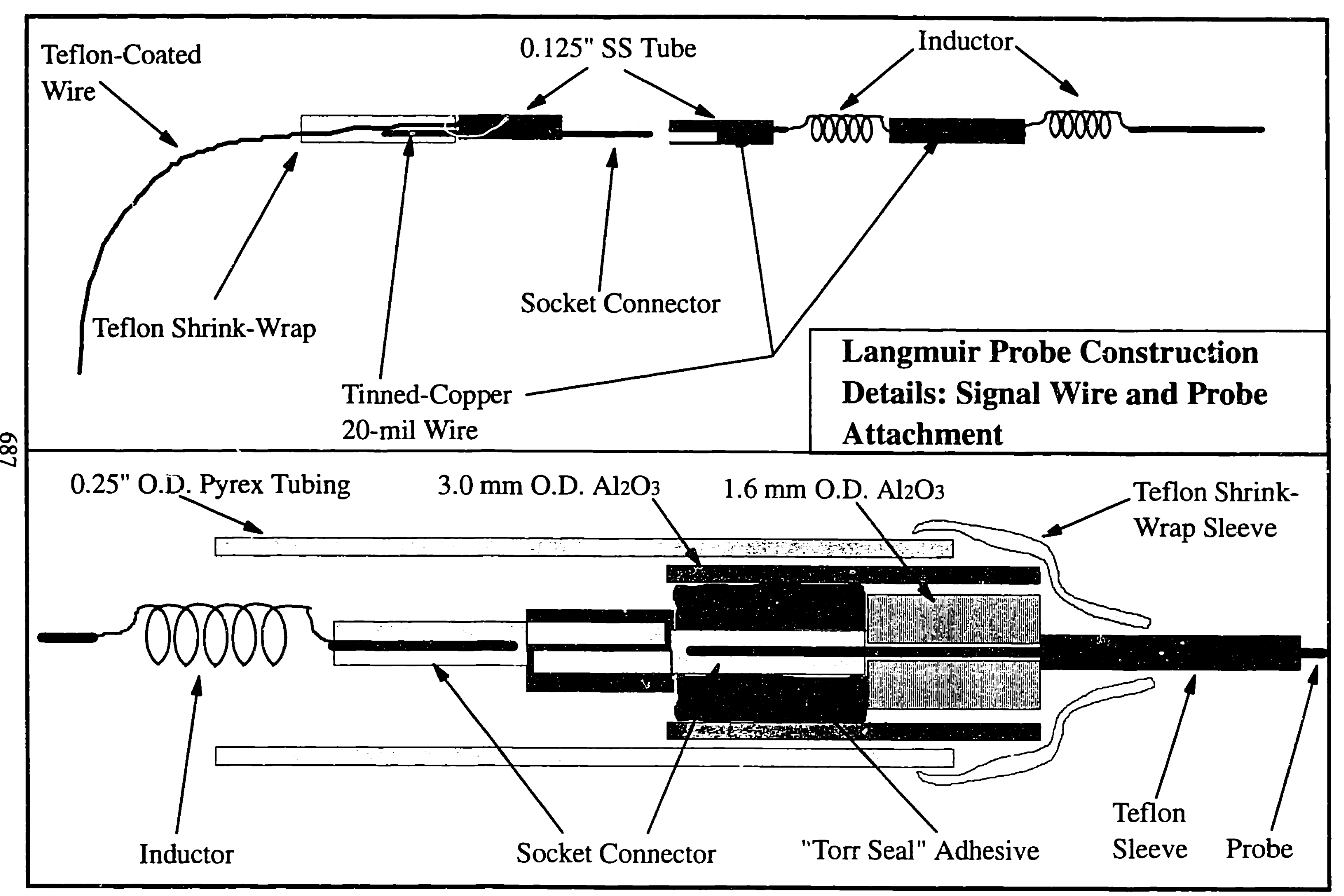
Model Number	Inductance (μH)
9230-68	100
9230-70	120
9230-72	150
9230-74	180
9230-76	220
9230-78	270

As mentioned in Chapter 7, the inductors were connected together by

(1) trimming the leads on the ends down to about 3-mm, and
(2) inserting the trimmed leads of 2 adjacent inductors into an approximately 3-mm long section of 0.125" SS tubing (available at Chem Lab Supplies in Bldg. 18) and then crimping them together. This provided for a solder-free connection of the inductors.

To removably attach wires and the tip to the probe assembly, a socket was used. It is available from Newark Electronics as Part # 89F5612. This socket accepts and holds a 20-mil wire, reuseably. The end of the socket was milled flat with the dremel and then inserted inside of a 0.125" SS tube with the final inductor inductor in the string. A section of the tinned-copper lead that was cut off of one of the inductors was attached to a section of Teflon®-covered wire by teflon shrink-wrap (Newark Electronics also) and the bared wire end and the tinned-Cu were crimped inside a piece of 0.125" SS tubing, leaving some of the tinned-Cu lead passing through for attachment to the inductors. (See figure on following page).

The probe tip was attached through the assembly shown in the figure on the following page. Two socket connectors were attached back to back with a 0.125" SS tube crimp. One of these connectors was located inside of two concentric Al_2O_3 tubes (Johnson Mathey Alpha Group, 3mm O.D. x 1.6 mm I.D. #31553 and 1.6 mm O.D. x 0.8 mm I.D. #31485) as shown in the figure. It was held in place with "torr seal". To one end of this fixture, the chain of inductors was attached. The other end of this passed through the 0.25" pyrex tube housing the inductors. The pyrex was fire-polished to reduce to I.D. slightly to make a snug fit to the 3.0 mm O.D. tube. The probe wire was inserted into this end of the assembly. A thin teflon sleeve covered the probe wire, except for the active tip. Teflon shrink-wrap was used to cover the junction of the pyrex tube and the alumina tube.



Teflon-Coated
Wire
0.125" SS Tube
Inductor
Socket Connector
Teflon Shrink-Wrap
Tinned-Copper
20-mil Wire
Langmuir Probe Construction
Details: Signal Wire and Probe
Attachment
0.25" O.D. Pyrex Tubing
3.0 mm O.D. Al2O3
1.6 mm O.D. Al2O3
Teflon Shrink-
Wrap Sleeve
Inductor
Socket Connector
"Torr Seal" Adhesive
Teflon
Sleeve
Probe