Statistical Metrology of Interlevel Dieletric Thickness

by

Tinaung Daniel Maung

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degrees of

Bachelor of Science in Electrical Science and Engineering

and

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1995

Author .....
Department of Electrical Engineering and Computer Science
January 27, 1995

Certified by .....
Duane Boning
Assistant Professor
Thesis Supervisor

Certified by .....
James Chung
Assistant Professor
Thesis Supervisor

Accepted by .....
F. R. Morgenthaler
Chairman, Departmental Committee on Graduate Students

Statistical Metrology of Interlevel Dieletric Thickness

by

Tinaung Daniel Maung

Submitted to the Department of Electrical Engineering and Computer Science
on January 27, 1995, in partial fulfillment of the
requirements for the degrees of
Bachelor of Science in Electrical Science and Engineering
and
Master of Engineering in Electrical Engineering and Computer Science

Abstract

Statistical metrology is a methodology for the systematic evaluation and quantification of variationin semiconductor manufacturing. This methodology applies a statistically significant number of electrical measurements on test structures designed for short loop process flows. Statistical metrology has been developed and applied to interlevel dieletric (ILD) thickness variation. A test mask for evaluating the ILD thickness has been designed using a two-level, half-factorial experimental design. Hence, for six factors, a total of thirty two test structures with various dimensions have been designed. In order to quantify the polysilicon and metal linewidth variation, resistive test structures with the same dimensions are placed in proximity to the capacitor structures. The variation extracted from these structures is used in conjunction with two-dimensional capacitance simulations to extract the thickness of the dielectric oxide. The data from these experiments is analyzed using statistical techniques and the main layout factors that affect the interlevel dielectric thickness are identified for dielectics planarized using BPSG reflow and chemical mechanical polishing (CMP) planarization techniques. Better comprehension of these variation sources improve the design and control of the ILD thickness in advanced interconnect technologies.

Thesis Supervisor: Duane Boning
Title: Assistant Professor of Electrical Engineering

Thesis Supervisor: James Chung
Title: Assistant Professor of Electrical Engineering

Acknowledgements

I would like to express my gratitude to Professor James Chung and Professor Duane Boning for their guidance and helpful criticisms throughout this project. Thanks to Eric Chang and Rajesh Devacha for their invaluable help with the testing and data analysis, to Robert Ha and Raj Sodhi for the prober programming, and to Susan Kim for the layout of the MIT mask. I would also like to acknowledge the assistance of many colleagues and staff at MIT Microsystems Technology Laboratory for the processing of the wafers. Especially helpful in this regard have been Quan Xiong, Jeffrey Kim, Ka-Shun Wong, Bernard Alamariu, and Pat Burkhart.

I thank Keh-Jeng Chang and Soo-Young Oh for their supervision and support at Hewlett-Packard through the course of my summer internship. The engineers and the staff at Hewlett-Packard provided a very supportive environment for undertaking the project. Dirk Bartelink, Gary Ray, Shisen Peng, Konrad Young and Crid Yu contributed to this project through helpful discussions. Lisa Martin, Hal Christiensen, Ed Lin, Fred Perner, Charles Weber, and Maria Yang provided the training and invaluable advice for the frame generation and mask layout. Kent Okasaki, Tom Eckstedt, and Mark Leonard were very helpful with the computer problems. Delores Gurreola, Don Bradbury, and Jeff Glenn assisted with wafer processing, and Glenn Fuller, Tony Hsu, Sandy Thomas, and Vijay Varma helped me out with programming in ICMS and testing of wafers.

Many thanks go out to my buddies at MIT. I appreciate all the good and bad times we have had together. Thank you for making my years here interesting, fun, and challenging.

Finally, a special note of thanks to my parents whose loving support and encouragement have been a source of strength throughout my life. They have sacrificed a great deal to ensure that I would have a better chance at success in life. I dedicate this thesis to my beloved parents.

This work has been sponsored in part by the Advanced Research Projects Agency under contract N00174-93-C-0035.

Contents

List of Figures

Chapter 1

Introduction

Process variation is a prevalent problem in any manufacturing process, and semiconductor manufacturing is no exception. The variation in semiconductor manufacturing can appear at the die level, wafer level, and wafer-lot level. Since the root causes of the variation at each level could be very different, one cannot assume that the variation in a smaller scope will be smaller than the variation in a larger scope. For example, the intradie variation could be larger than wafer to wafer variation, and the wafer to wafer variation could be larger than lot to lot variation depending on the nature of the causes of the variation. We can have very good wafer to wafer control in a chemical vapor deposition process while the intradie variation could be large due to the pattern dependencies of the deposition process. Each level of variation demands its own control methods and we need to understand the causes at each level.

Moreover, variations in semiconductor processing involve both deterministic and random components. The systematic component of the variation has physical cause and could be controlled by manipulating process and equipment parameters. However, if the nature of the variation is not well understood, the deterministic component might be lumped into the random component and valuable process information could be lost. Hence, it is very important to be able to sort out these components by understanding the sources of the variation.

1.1 Statistical Metrology

Statistical metrology is an approach to quantify the variation and identify the sources of variations in semiconductor processing. It utilizes a significant number of electrical test structures that are sensitive to the systematic process variations associated with pattern, spatial, and process dependencies. Short loop process flows are used because the number of electrically assignable effects decreases as the wafers undergo more process steps. Moreover, by using short loop process flows, rapid feedback of results could be provided to the process engineers.

For the design of test structures, the statistical design of experiments is used in order to obtain the maximum amount of information from the experiments. We also make extensive use of TCAD structure/device simulation tools in conjunction with the electrical measurements.

Since electrical test structures must be fabricated through several steps in a process sequence, these "short-loop" process flows contain a number of process modules. Hence, the electrical data collected from these test structures is a confounded sum of the systematic and random contributions from each of the process steps. Statistical techniques can then be used to derive the individual contributions of each of the process steps to the overall parameters.

Once the sources of variability are identified and quantified, appropriate steps could be taken to model the variability and control the parameters involved. This information could help the process engineers in their process calibration and equipment purchasing decisions. Circuit designers could also make use of these statistical models by incorporating them into circuit design rules. The concepts of statistical metrology can be applied to many areas in semiconductor fabrication such as lithography, etch, and thin-film deposition. Recently, researchers have examined the variation in polysilicon critical dimension [24][6]. For this thesis, we focus our efforts on developing a statistical metrology methodology for interlevel dielectric thickness variation.

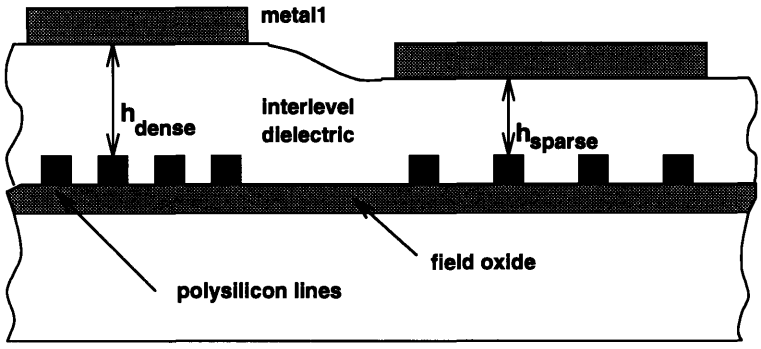

Figure 1-1: Cross-sectional View of a Multi-level Metal Process

1.2 Interlevel Dielectric Thickness

The interlevel dielectric (ILD) is an insulating dielectric layer that separates one layer of conductor from another as shown in Figure 1-1. The ILD is usually a variation of silicon dioxide deposited using a chemical vapor deposition process.

Today's circuit performance is increasingly limited by interconnects rather than device performance since the interconnect dimensions do not scale well with transistor size. The problem of parasitic resistance and capacitance associated with the metalization is becoming worse as interconnects have to be scaled down to achieve larger chip densities. Circuit designers usually model the interconnect as a lumped RC line where the load capacitance C is dominated by the parallel-plate capacitance between the conductors in adjacent metal/polysilicon layers [25]. These adjacent conducting planes are separated by the thickness of the ILD which is inversely proportional to the parallel plate capacitance. As a result, the ILD thickness is one of the most important parameters that directly impacts the parasitic interconnect capacitance.

The ILD thickness is believed to be a function of both layout factors and processing factors. Any variation in the ILD thickness due to various factors results in interconnect capacitance variation which can negatively impact circuit performance. For example, a pair of identical clock lines running over different topographies could result in clock skews at the end of the lines due to the difference in parasitic capacitance created by ILD thickness variation. Since clock skews usually translate to loss

of circuit speed, circuit designers would like to see ILD thicknesses that are uniform across the die, wafer, and lot level.

The need for planar dielectrics is also being driven by lithographic requirements. Current state-of-the-art photolithography tools require depths of focus less than 0.5μm. Thus, the distance between the highest point and the lowest point on the die must be less than 0.5μm in order for these lithography systems to be able to focus across the entire die. The variations in resist thickness and the tool's focusing errors further decrease the error budget due to topography variations [9].

Moreover, with today's VLSI technologies adding more and more metal layers to the interconnection schemes, the need for a high degree of planarization is increased. The surface of a wafer must be planarized at each level in order to prevent the topography roughness from growing with each level. Failure to planarize the surfaces adequately could result in unacceptably low circuit yields due to problems such as poor step coverage of metal lines [10].

1.2.1 ILD Planarization Techniques

To resolve the dielectric planarization problem, many techniques and dielectric materials have been explored. Such techniques include oxide reflow, resist etchback, spin-on-glass with etchback, deposition-etch-deposition sequences, and electron cyclotron resonance (ECR) oxide deposition [10]. While these techniques provide degrees of smoothing, they are quite limited in that level-to-level flatness over an entire die or wafer demanded by today's circuits cannot be achieved with these techniques. The best emerging candidate for global planarization seems to be a technique called chemical mechanical polishing (CMP). In this thesis, a comparison will be made between ILD planarizations using reflowed BPSG and chemical mechanical polishing (CMP); these technologies are described below.

Reflowed BPSG

BPSG has been one of the most widely used materials for ILD. Reflowed BPSG has been an attractive planarization method due to its simplicity, affordability, and high

throughput.

BPSG glass is an amorphous mixture of silicon, boron and phosphorus oxidized to a stoichiometry of SiO_2, B_2O_3, and P_2O_5 [20]. It is deposited by the controlled reaction of silane, phosphine, diborane, and oxygen. By reflowing the glass at high temperatures (typically over 800°C), more planar surface with smoothing over the vertical steps can be achieved. The ability of the BPSG to reflow depends on the phosphorus and boron concentration in the glass. Because the reflow of BPSG requires high temperatures, it cannot be used as an ILD for higher-level metal layers. Temperatures much above 400°C cause hillocks and electromigration problems in aluminum [9]. In this thesis research, BPSG is used due to its availability at MIT and the short loop nature of the process.

Chemical Mechanical Polishing (CMP)

Recently, CMP has emerged as a promising planarization technology. Local planarization can be handled by gap filling techniques, but extreme planarity requirements on the global scale can be met only by chemical mechanical polishing (CMP) [18].

Chemical mechanical polishing involves removal of a sacrificial ILD layer to obtain a high degree of planarity. It is inherently a nonlocal process and produces planarization over areas covering the stepper field width. Removal of the oxide layer takes place through a combination of mechanical and chemical action, using a polyurethane pad and abrasive slurry dispersed in alkaline solution [19]. The wafers are held against a rotating polishing pad wet by a slurry consisting of colloidal silicon dioxide at a high pH. Removal rate in CMP depends greatly on pattern density, local geometry, and point-to-point temperature and pressure variations. By relieving the severity of the ILD topography, CMP is also known to reduce the defect density due to broken metal lines.

1.3 Motivation

The escalating need for ILD thickness uniformity is driven by the push to smaller geometries and higher chip speeds. Semiconductor manufacturers today are moving towards larger wafer and die sizes, and they are encountering increasing problems due to variation. In addition, today's high speed chips demand extremely tight tolerance limits, and the nonuniformities in fabrication are becoming a significant fraction of the total error budget. A chip that works at a lower frequency may not function properly at full speed due to the limitations imposed by variation. Since we cannot expect to fix what we cannot measure, it is very important to quantify the variations and identify the sources of these variations.

The ILD thickness variation within a die is believed to be due in part to the pattern variations in underlying structures whereas the wafer to wafer and lot to lot variation is believed to be due to poor end point detection and equipment uniformity and control. However, it is not well understood how the geometry of the structures affect the ILD thickness. The effects of the underlying structures on the ILD variation need to be better understood in order to give the circuit designers more accurate specifications which will be used in creating SPICE circuit models needed for determining the interconnect propagation delays, clock skews, etc. This understanding will also help process engineers calibrate the equipment to achieve better control of the ILD planarization process. The results of statistical metrology will also provide the data necessary for TCAD tool calibration.

In this thesis, we investigate the primary layout and spatial factors that affect the ILD thickness. We develop a methodology that includes electrical test structure design, automated data collection, and statistical data analysis. This methodology is exercised for two different ILD planarization techniques, namely reflowed BPSG and chemical mechanical polishing (CMP). The "MIT" process uses reflowed BPSG as ILD whereas the "HP" process uses CMP for ILD planarization.

Chapter 2 discusses the step by step methodology for statistical metrology. We begin by considering the issues involved in the design of the factorial experiment.

The experimental factors that are chosen for the two-level factorial experiment are described. We look at the test structure design for the capacitor structures along with the resistive linewidth structures and sheet-resistance structures. The overall strategy for the modules and mask design is presented for both the MIT and HP masks. The MIT and HP process flows for fabrication of the test wafers are presented. We then describe the simulation issues and the methodology followed for converting the measured capacitance into ILD thickness. A description of the equipment setup, test program algorithm, and data collection procedure is also given. Chapter 3 contains the data analysis for both MIT and HP wafers. We present the spatial dependency and layout factor dependency for the MIT and HP wafers. The results from Analysis of Variance (ANOVA) are also described. Data analysis done on the area-intensive structures is presented separately. Chapter 3 also discusses the limitation of the current data analysis. Chapter 4 summarizes this work and gives suggestions for future research.

Chapter 2

Methodology

Material characterization techniques such as scanning electron microscopy (SEM) and Transmission electron microscopy (TEM) have been very widely used in semiconductor metrology. While these material techniques are very precise, they are prohibitively expensive for obtaining statistically large amounts of data. Alternative techniques such as optical film thickness measurements require large unpatterned areas and cannot be taken reliably over patterned materials such as metal and polysilicon lines.

The advantage of electrical test structures is that a large amount of data can be collected in a relatively short time. Moreover, the data collection can be automated so the profile of the samples can be easily determined. Our methodology involves constructing a design of experiment using a set of layout factors and producing a test mask. We then extract the ILD thickness information from a combination of capacitance and resistive linewidth measurements with TCAD simulations. This methodology is followed for both the MIT process using reflowed BPSG as ILD and the HP process using chemical-mechanically polished TEOS as ILD.

2.1 Experimental Design

The MIT experiment involves two phases. The first phase is a screening experiment which is used to identify the most important factors and interactions that affect the ILD thickness. The second phase of the experiment is planned to involve more levels of

these chosen factors which would give us a more accurate model for the ILD thickness as a function of the chosen layout and process factors.

2.1.1 Choosing Design Factors

The objective of the first pass experiment is to identify the factors that impact the variability of ILD thickness, so the selection of the design factors is one of the most important decisions. The linewidth and spacing of the lines in the test structures are easily the top candidates because there has been much research done on the impact of these parameters on the ILD thickness. We also add the geometric orientation of the structure as a factor because the gas flow at the wafer surface in the dielectric deposition could be influenced by the orientation of the lines.

One of the criteria for selecting the design factors is that they be physically independent of each other. Hence, area of the test structure, which depends on the linewidth, spacing, length, and number of lines, is left out. The length of the lines and the number of lines, which are independently controllable quantities, are added as design factors. Since we also want to assess the impact of neighboring structures, the presence or absence of an interaction ring around the structure is also included as one of our design factors. The design factors are shown in Figure 2-2.

2.1.2 Factorial Experiment Design

A two-level full factorial experiment is originally considered for the screening experiment, but due to inadequate chip space for 64 test structures, an alternative design using a half-factorial of a 2^6 factorial design is used. By choosing a half-factorial experiment, a design resolution of VI is achieved. Hence, the main effects are confounded with five-factor interactions, the two-factor interactions are confounded with four-factor interactions, and so on [1]. This resolution is considered to be more than adequate for our experiment because we expect relatively insignificant fourth order or higher interactions.

A half-factorial design as shown in table 1 is constructed. A full 2^5 design is written

for the first five factors: line width, line spacing, number of fingers, finger length, and geometric orientation. The column of signs for the product lw*ls*nof*fl*go is used to define the levels for interaction ring. In fact, any combinations of the first five factor levels could be used for the sixth factor, but a half factorial design with maximum possible resolution is achieved with the choice made above.

If needed, the other half-fraction could be added subsequently to make this design into full-factorial experiment with six factors. Moreover, if any one of the factors is later found to have no effect on the result, we are left with a complete factorial experiment in the five remaining factors regardless of which factors they are [1].

2.1.3 Choosing Factor Levels

Since this is a two level screening experiment, the high and low factor levels must be chosen with enough spread in order to capture the significance of each factor. The finger width dimensions are chosen to be 1.5μm and 5 μm since 1.5μm is the minimum design rule for polysilicon lines in the MIT facility. Likewise, the line spacings are chosen to be 2μm and 4μm. The number of fingers for the low level is 50 so that the capacitance of the smallest test structure is above 1pF. The high level for the number of fingers is 100. For the geometric orientation factor, half of the capacitors are horizontally oriented while the other half are vertically oriented.

The capacitance for the structures are estimated using the parallel plate capacitance formula $\epsilon A/d$ with the area being the product of the number of fingers, linewidth, and the length of a finger. Since the capacitance due to fringing fields is not included in the estimate, this is a conservative estimate. The measured capacitance from the test structures will be greater than the estimates due to fringing fields, pad capacitance, and parasitics from the test equipment.

Since we want to make the mask usable for conducting layers up to metal 4 at HP, the factor levels for the HP structures are chosen so that the minimum design rules are met up to metal 4. The additional chip space available for the HP chip also gives us some flexibility as to the range of the layout factors we can vary. Since the HP process is designed for a thicker dielectric, the HP test structures need a larger

Structure	Linewidth	Spacing	Length	# Fingers	Orientation	Interaction
1	-	-	-	-	+	+
2	-	-	-	+	-	+
3	-	-	+	-	-	+
4	-	-	+	+	+	+
5	-	+	-	-	-	+
6	-	+	-	+	+	+
7	-	+	+	-	+	+
8	-	+	+	+	-	+
9	+	-	-	-	-	+
10	+	-	-	+	+	+
11	+	-	+	-	+	+
12	+	-	+	+	-	+
13	+	+	-	-	+	+
14	+	+	-	+	-	+
15	+	+	+	-	-	+
16	+	+	+	+	+	+
17	-	-	+	+	-	-
18	+	-	+	-	-	-
19	+	-	+	+	+	-
20	-	-	+	-	+	-
21	+	+	+	+	-	-
22	-	+	+	-	-	-
23	-	+	+	+	+	-
24	+	+	+	-	+	-
25	+	-	-	+	-	-
26	-	-	-	-	-	-
27	-	-	-	+	+	-
28	+	-	-	-	+	-
29	-	+	-	+	-	-
30	+	+	-	-	-	-
31	+	+	-	+	+	-
32	-	+	-	-	+	-

Table 2.1: The Factorial Experiment Design

minimum area than the MIT test structures. However, this need is partially offset by the finer geometries available for the HP process. The detailed description of the factor levels for the HP experiment are not disclosed due to their proprietary nature.

2.2 Test Structure Design (MIT/ HP)

The test structures for this experiment are designed for automatic electrical measurements using a wafer prober. They are designed with the probe pads as an integral part of the structure. The probe pads are in a 2 by 12 array as shown in Figure 2-1. The test structures are electrically isolated from each other in order to avoid parasitics that could be caused by sharing of common conductors. Moreover, the mask is laid out in a modular pattern so that all the test structures are accessible with one standard probe card.

2.2.1 Capacitor Structures

The basic test structure for the experiment is a capacitor with an edge-connected conductor as the bottom electrode and a large area blanket conductor as the top electrode. The top electrode encompasses the bottom electrode. The connecting edge bars for the lower electrodes are 10μm wide.

2.2.2 Van der Pauw Structures

Van der Pauw structures are added to each test module in order to measure local sheet resistance. Since the sheet resistance can vary significantly over the die, it is important that the Van der Pauw structures are close to the linewidth structures.

2.2.3 Linewidth Structures

Linewidth structures are basically four point resistors. A linewidth structure is added next to each capacitor in order to estimate the linewidth variation of the lines inside the capacitor. Each linewidth structure has a design width identical to the linewidth

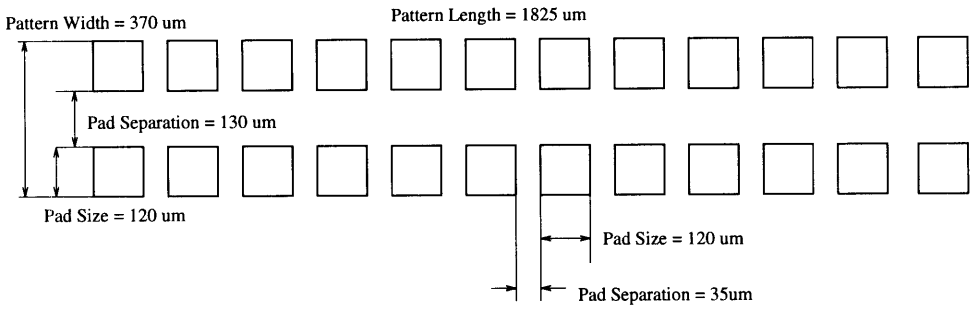

Figure 2-1: Probe Pad Pattern

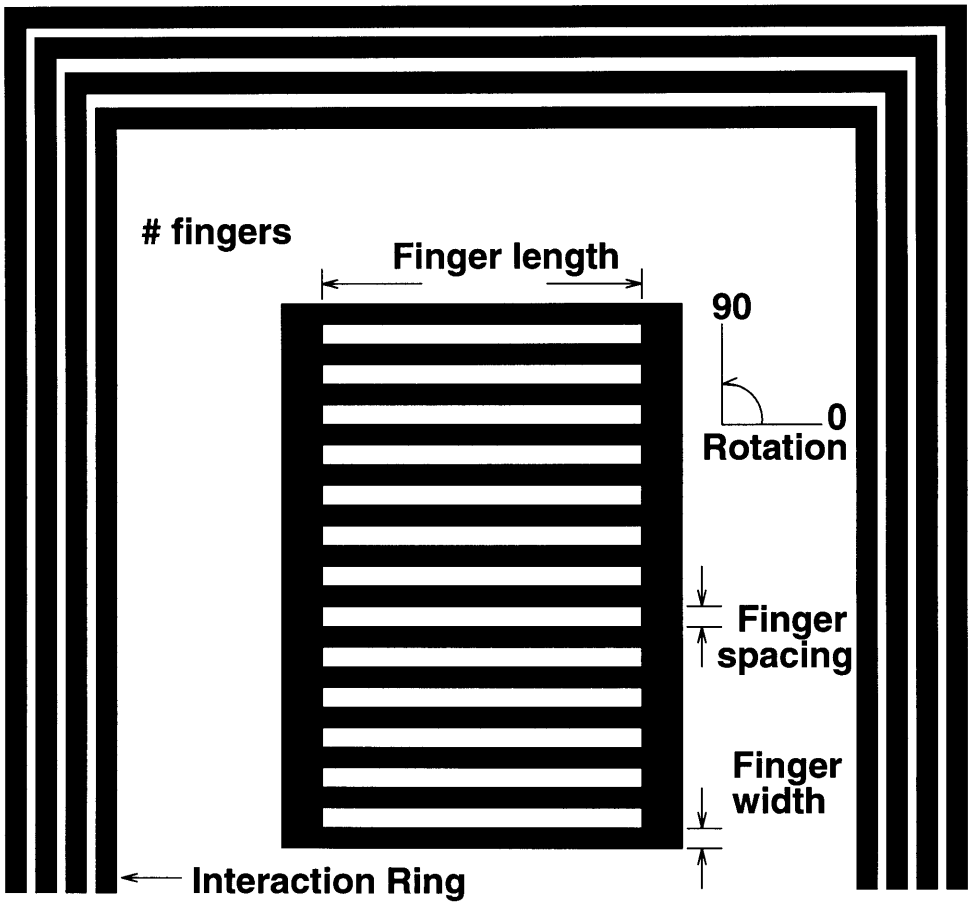

Figure 2-2: Top View of a Capacitor Structure

of the lines in the capacitor next to it. It is assumed that if the lines are close enough to each other, the linewidth variation due to photolithography and etch steps would be very similar. This assumption is verified using independent physical analysis from SEM pictures. The polysilicon linewidths are known to depend on the lithographic pattern [24]. Hence, in order to mimic the pattern of the capacitor structures, each of the linewidth structures is surrounded by dummy lines although the resistance measurement is taken only on the single line in the middle of each structure.

2.3 Subdie (Module) Design

A subdie or module is defined as a collection of devices that can be simultaneously contacted by a probe card's pin array. In the case of our probe card, we have a 12 by 2 pin array on our probe card so we have the capability to make 24 connections for each module. Hence, our module consists of a 12 by 2 array of probe pads overlaying the test structures. While it is possible for us to layout the test structures arbitrarily inside each module, we follow a regular pattern of devices with four capacitors and four resistive linewidth structures corresponding to each of the capacitors in order to ease the task of writing the test program. A Van der Pauw structure is placed in the middle of each module to measure the local sheet resistance with the assumption that the sheet resistance variation within a module is minimal.

2.4 Mask Layout

2.4.1 The MIT Mask

The MIT test mask is laid out using the Berkeley KIC program. The mask size is the MTL standard 1cm x 1cm. A modified subset of the standard drop-in pattern cells for the MTL baseline process is incorporated into the mask. This subset of test patterns includes verniers, optical patterns, wafer alignment crosses, and a contact chain. The verniers are used for quantifying the misalignment between two layers during a given photolithographic step. The optical patterns enable the mask making

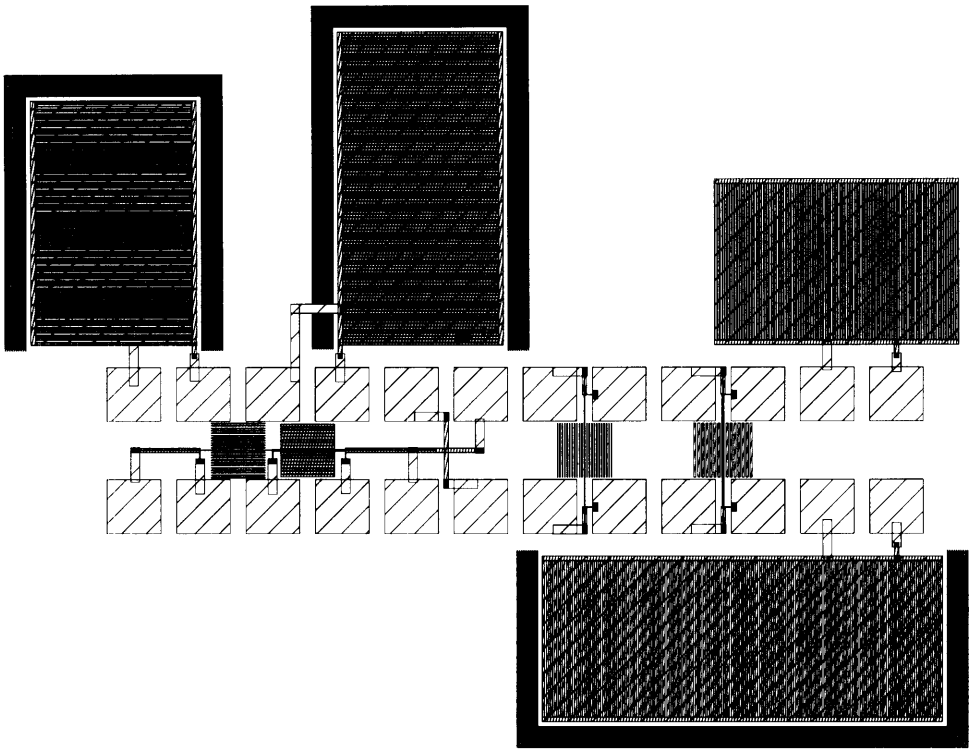

Figure 2-3: Top View of a Subdie / Module

equipment operator to standardize the mask development process and they need to be included in every mask layer. The wafer alignment marks have to be placed at known coordinates and the information about these coordinates and the stepping distance are used to expose each level and to align other layers to this level [13]. The contact chain can be used to evaluate the opening of contacts during etch time and to measure the contact resistance at the end of the process.

Each test structure is replicated three times on each die. Duplicates provide redundancy as well as information regarding the spatial variation of ILD thickness over each identical structure. The duplicates are pseudo-randomly distributed on the die in order to get a good spatial mapping. Care is taken to ensure that each structure is placed near the edges as well as at the interior parts the die.

An array of blanket capacitors are also added to the bottom of the die. These capacitors are to be used for measuring the dielectric constant of the ILD which could vary from wafer lot to lot depending on the dopant concentrations in the dielectric. We

also add an array of dummy pads which are used to measure the parasitic capacitance between the pads and the chuck.

The resulting mask design is shown in Figure 2-4. The die consists of six rows of test structures that are four modules wide. The minimum distance between the modules is kept at 50μm to minimize interaction between adjacent modules. Each wafer contains 52 of these die.

2.4.2 The HP Mask

The HP test mask is laid out in HP's ChipBuster IC design software. Because we want to make the mask usable for conducting layers up to metal 4, the structure dimensions are chosen such that design rules for metal 4 are met. The HP die size is significantly larger than the MIT die size so we are able to fit four duplicates of each test structure on each die. The die is divided into four quadrants and the first three quadrants of the die are allocated for the test structures that are similar to the MIT test structures.

Some modifications are needed for the MIT mask in order to optimize the HP mask for chemical mechanical polishing. We anticipate the CMP process to be area dependent because of CMP's global scope. Hence, a quarter of the die is allocated for area intensive test structures. We also expect that proximity effects could play a large role in the CMP process so we design half of the area intensive structures with interaction rings that are 100μm wide and 10μm away from the test structures. The other half of the test structures are designed without the interaction rings. The sizes of the area intensive capacitors range from 300μm x 300μm to 1000μm x 1000μm. There are a total of eight different capacitor configurations on the fourth quadrant of the die, and each capacitor is replicated three times. Each of the capacitors also have a linewidth structure adjacent to it.

The area intensive structures are separated from the original test structures for a number of reasons. First of all, area is not an independent parameter from the other layout factors so including area as a factor in the original design of experiments would create undesirable confoundings in our design. Secondly, the area intensive structures

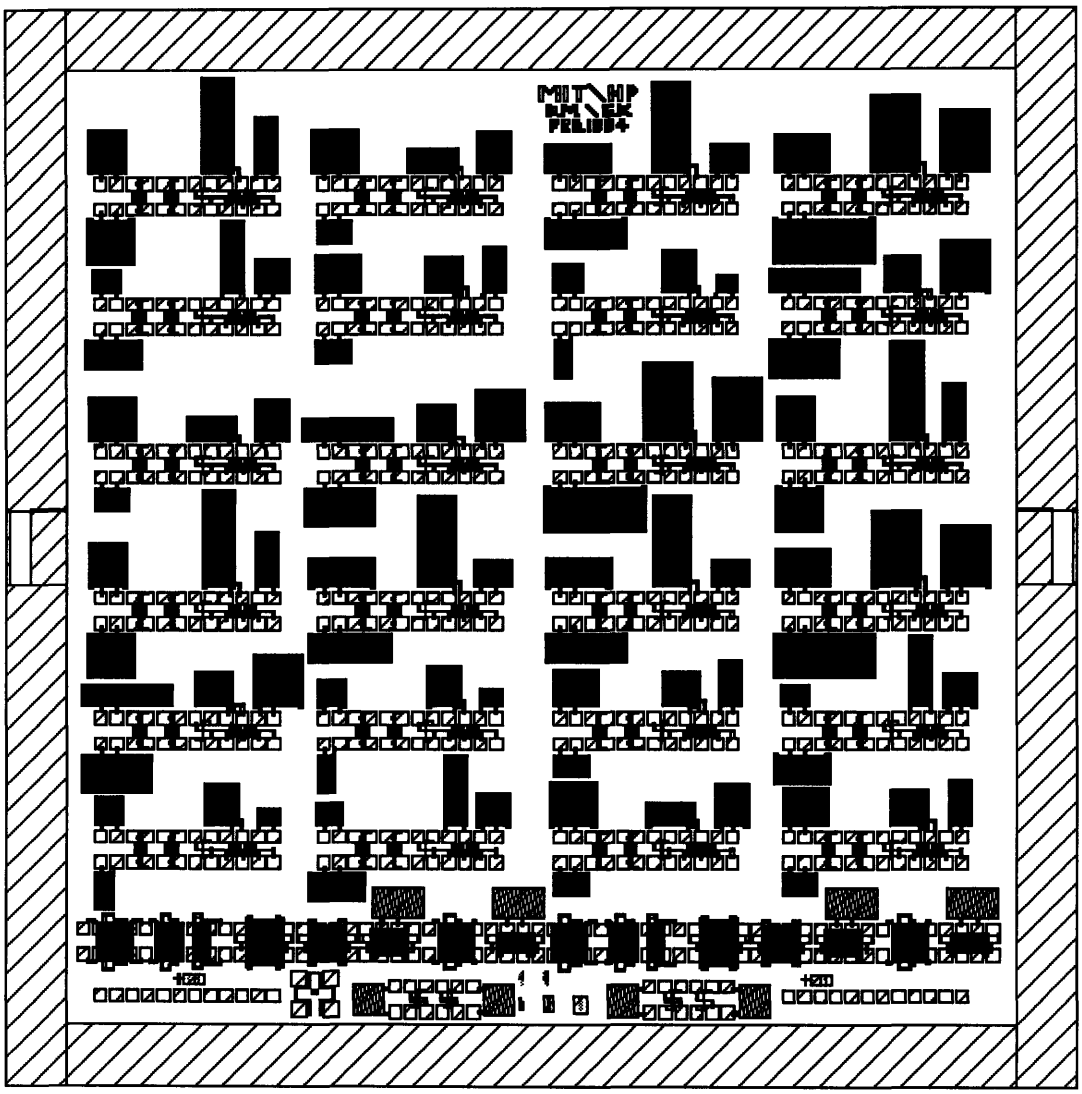

Figure 2-4: The MIT Test Mask

have to be quite large in order for them to significantly affect the CMP process, and incorporating these large area structures in the original design of experiments would require a much larger chip size than the one that is available to us. Figure 2-5 shows a top view of the HP mask.

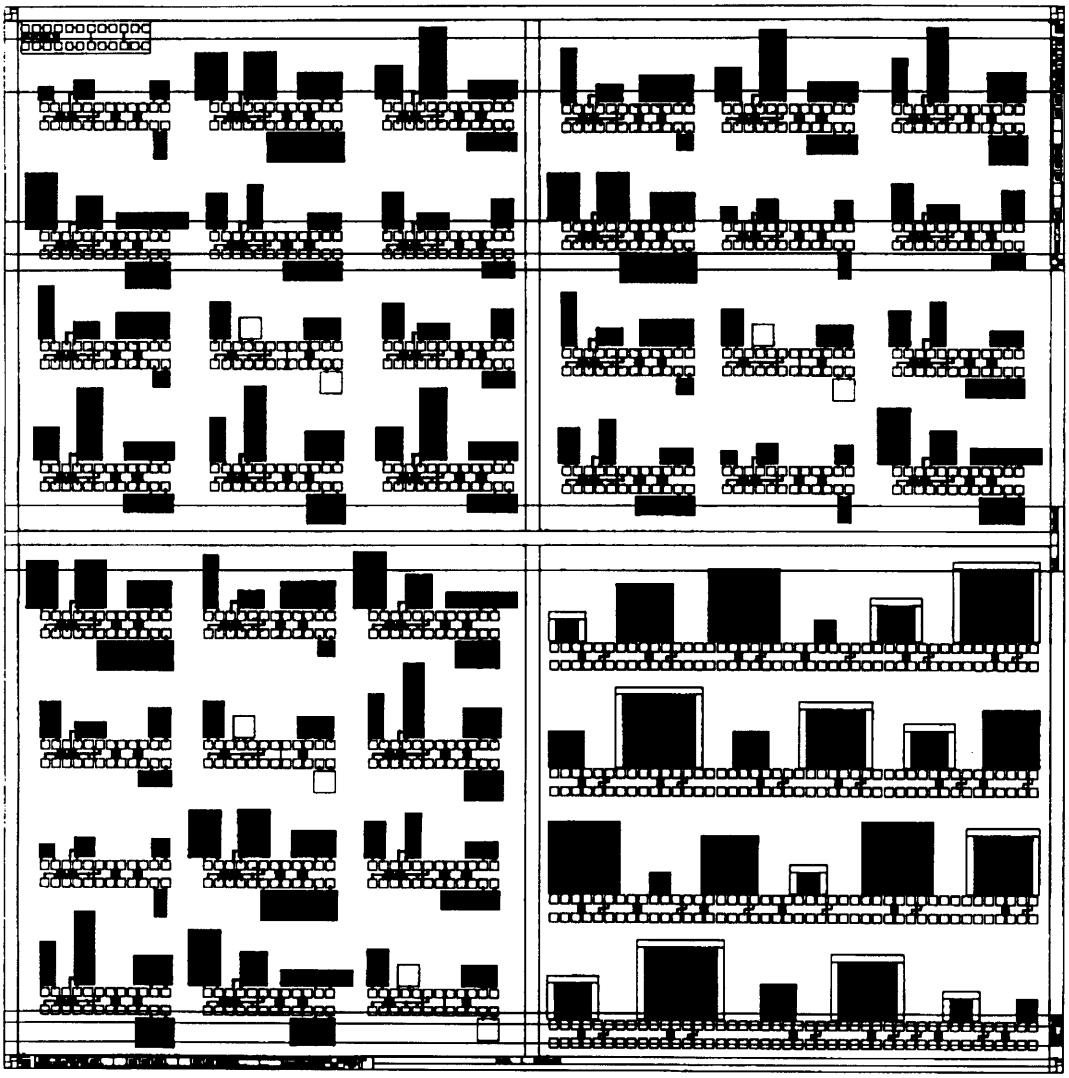

Figure 2-5: The HP Test Mask

2.5 Process Flow

The process flow is designed to include a minimum number of steps in order to minimize the number of confounding factors and to provide rapid feedback of data. A process split in the interlevel thickness deposition is included as an additional experimental factor in order to determine how the layout dependencies of ILD thickness correlate to process factors.

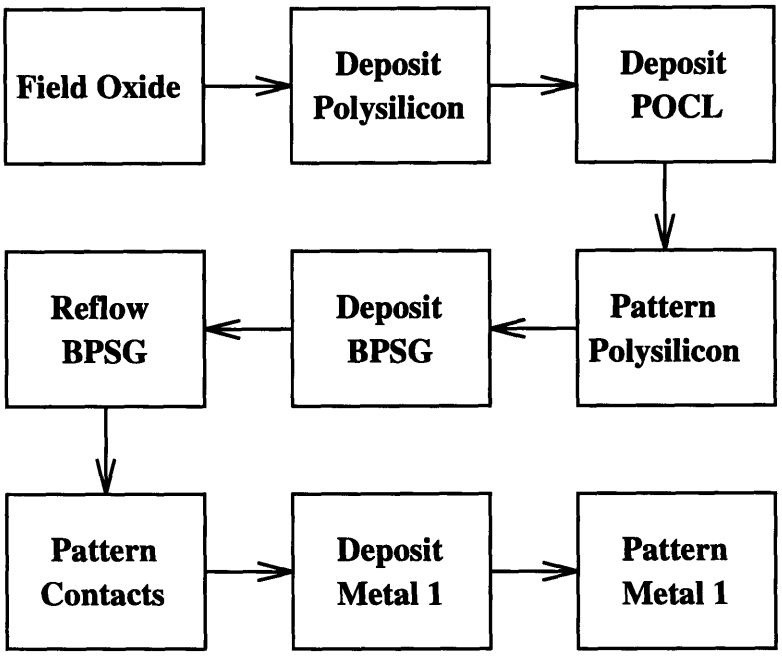

Figure 2-6: The MIT Process Flow

Microsystems Technology Laboratory's CMOS baseline recipes are used with minor modifications to fabricate the test structures. MTL's CAFE (Computer Aided Fabrication Environment) and PFR (Process Flow Representation) are used for the fabrication process. Twelve p-prime wafers are used as the starting substrate material. Figure 2-6 shows the process flow for our first run. The field oxidation is done at 950°C for 5 hours for a target thickness of 5000±300Å. Polycrystalline silicon is then deposited for a target thickness of 7000Å. In order to get a large conductivity for the polysilicon, a large dose of POCL is deposited. The POCL deposition results in a layer of phosphorous glass on top of polysilicon which must be etched back. The first

photolithography step is done using the polysilicon mask. Special care is needed for this step since this mask contains the finest dimensions among all the photolithography steps. The polysilicon is then etched in plasma using a standard CMOS baseline recipe. The polysilicon structures are manually inspected for possible defects using a microscope.

The interlevel dielectric used in the MIT process is BPSG or BoroPhosphoSilicate Glass which contains 4% boron and 4% phosphorous. It is deposited on top of patterned polysilicon at 400°C. A process split is made at this step. The target thickness of BPSG is 5000Å for half the wafers and 6000Å for the other half. For planarization, the BPSG glass is reflowed in a separate furnace at 925°C for 15 minutes. Contact windows are next opened in the oxide layer to allow electrical connections to be made between metal 1 and polysilicon. An aluminum layer is deposited on top of the BPSG layer after the contact etch, the backside polysilicon etch, and the backside oxide etch. Since the contact hole sizes are much larger than the thickness of the ILD, good step coverage is achieved using a CVD process. The metal layer is then patterned using the third level mask. The wafers are sintered subsequently. Sintering allows any interface layer that exists between the aluminum and the ILD to be consumed by a chemical reaction, and to allow the metal and polysilicon to come into intimate contact through interdiffusion [10]. The sinter step is carried out in a diffusion furnace at 400°C for 30 minutes in the presence of a forming gas.

During each of the deposition steps, the wafer orientations and the wafer order in the boat are randomized in order to eliminate the systematic bias due to orientation.

Figure 2-7 shows the process flow used at HP which is quite different from the MIT process. For ILD, we use TEOS material planarized by chemical mechanical polishing. Metal level 1 and metal level 2 are used as conducting layers instead of polysilicon and metal 1. Since HP's contact dimensions are smaller, they have to be filled with tungsten plugs with the aid of a TiN glue layer.

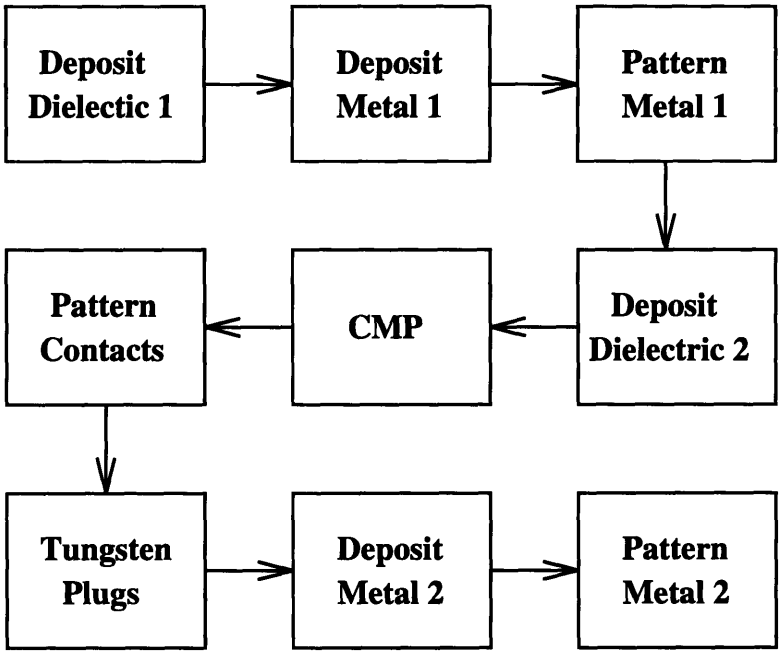

Figure 2-7: The HP Process Flow

2.6 Simulation

2.6.1 Simulation Issues

Capacitance simulations are needed for conversion of measured capacitance to inter-level dielectric thickness. The Raphael simulation program is used for capacitance simulations [4]. One of the main considerations is the use of 2D vs. 3D simulations. While the 3D simulations are more accurate because they take into account the fringing fields at the ends of conductor lines, the huge amount of computation time they consume is impractical for the hundreds of simulations required. Hence, we elect to perform multiple 2D simulations to approximate the 3D structure.

The bridging sidebars for the capacitors were omitted from the initial simulations, but we later determined that the addition of sidebar capacitance significantly alters the results so they are later also considered in the simulation. The sidebars are simulated separately from the middle conductor lines and the resulting capacitance is added to the total capacitance. This step is done inside the capacitance to thickness

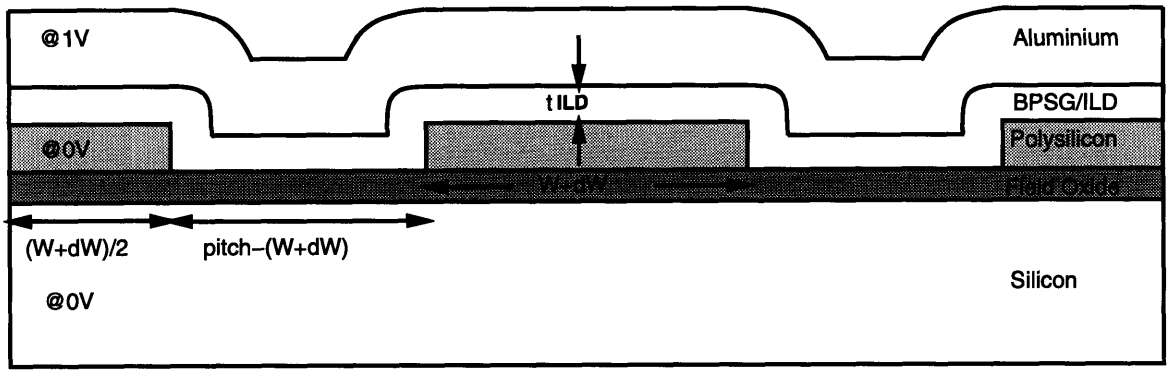

Figure 2-8: Cross-sectional View of a Test Structure Used for Simulation (BPSG Reflow)

conversion program.

2.6.2 Simulation Test Structure

Due to the computationally intensive nature of the simulations, it is not practical to simulate the whole test structure. It must be broken down into a smaller structure representative of each of the lines in the capacitor. The capacitance for the entire capacitor is then extrapolated from the basic test structure.

Figure 2-8 shows a cross-sectional view of the structure used for simulation. It consists of a full-width conductor line sandwiched between two half-width conductor lines. We apply +1V to the top conductor plate and the conductor lines are held at ground. Since the charge is the product of capacitance and the voltage, the capacitance per unit length in this case is equal in magnitude to the charge per unit length given by the simulator. A planar dielectric model was originally used for the original structure assuming that reflowed BPSG would give fairly good local planarization. However, SEM analysis later showed that the BPSG deposition is very conformal so the test structure for the simulation had to be changed to reflect the physical structure. Figure 2-9 shows the SEM cross-section of a typical structure.

The chemical mechanical polished wafers for the HP process, however, are expected to have much more planar ILDs and the original test structure shown in Figure 2-10 is used for the simulation of test structures for these wafers.

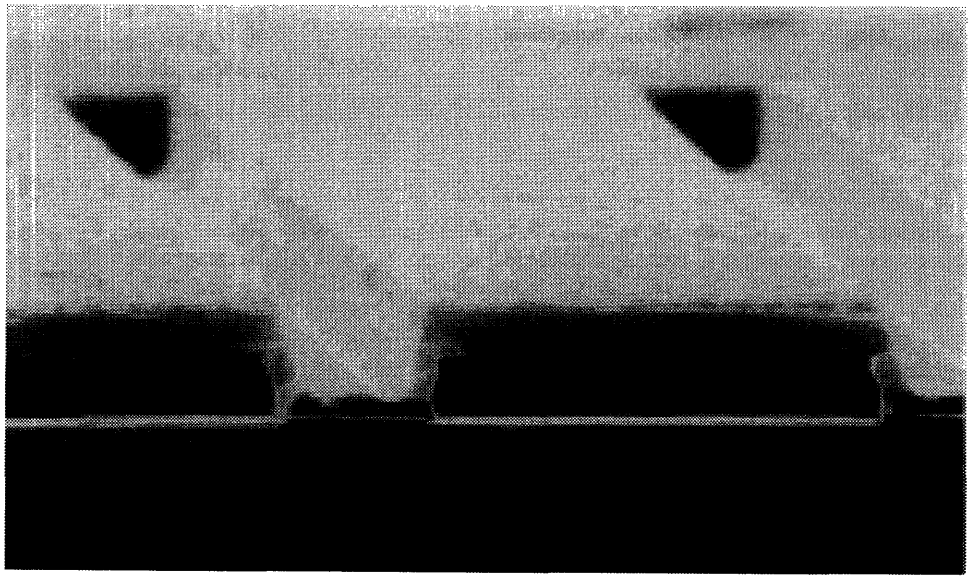

Figure 2-9: SEM Cross-section of a Test Structure

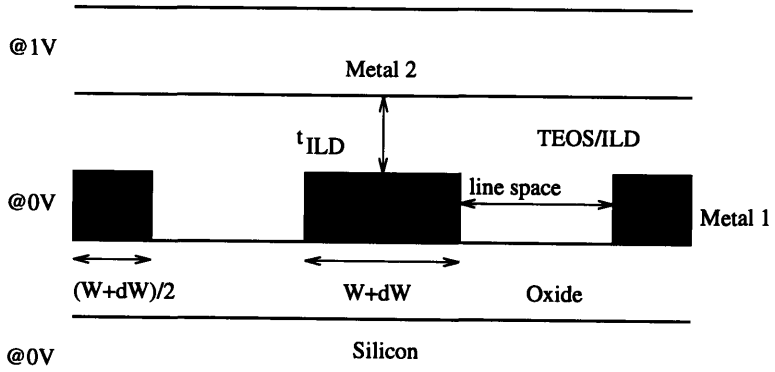

Figure 2-10: Cross-sectional View of a Test Structure Used for Simulation (CMP)

2.6.3 Interlevel Dielectric Thickness Estimation

For the ILD thickness estimation, we extend the methodology of [5] (Chang et al.) as shown in Figure 2-11. Capacitance simulations are performed for each unique drawn linewidth/spacing combination in the experimental design. By changing linewidth variation values in 0.05μm steps, and changing the ILD thickness value in 0.01μm steps for each linewidth variation value, a family of capacitance vs. ILD thickness curves as shown in Figure 2-12 is generated. We keep the pitch (linewidth + spacing) constant for all simulations. The ILD thickness is then estimated via two-dimensional linear interpolation for any given measured capacitance and linewidth value. This whole process of ILD thickness extraction from the measured linewidths and capacitances is automated using a combination of C programs and UNIX shell scripts.

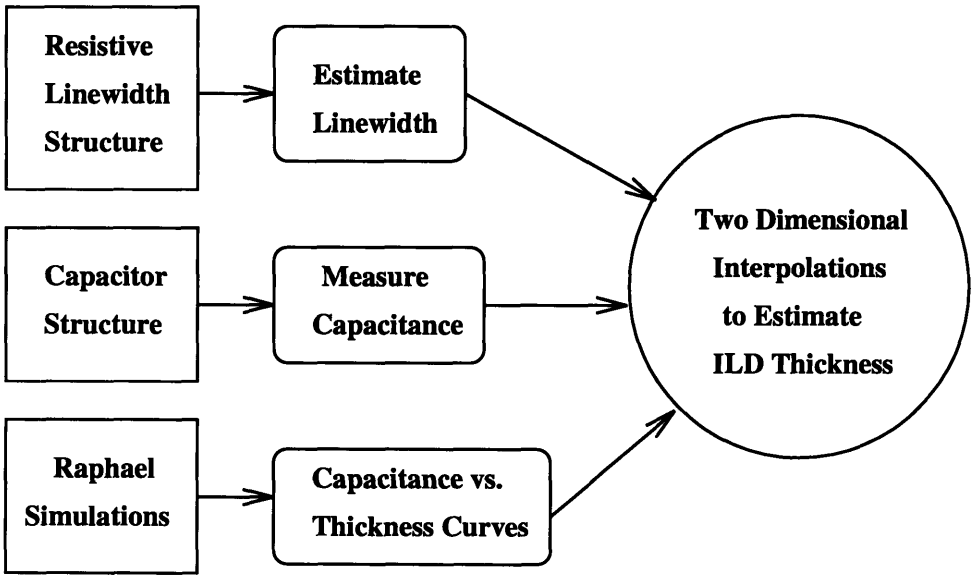

Figure 2-11: ILD Thickness Estimation Methodology

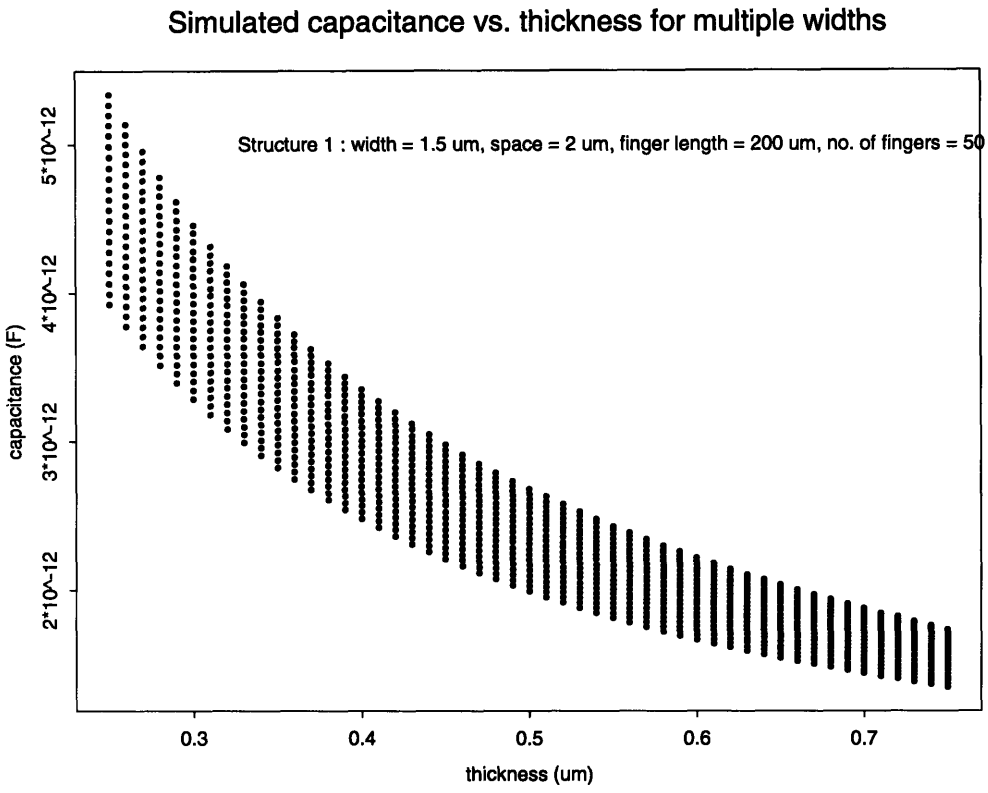

Figure 2-12: Measured Capacitance vs. ILD Thickness Curves

2.7 Testing

2.7.1 Equipment Setup

The equipment setup at MIT includes an HP4062B Semiconductor Parametric Tester, an R&K 1032 wafer prober, and an HP3000 computer for running the test program. The HP4062B contains four instruments: a 4141B DC measurement subsystem, a 4085A switching matrix, a 4084B switching matrix controller, and a 4280A capacitance measurement subsystem. The connections between the source measurement units and the probe card pins are made via a switching matrix unit.

2.7.2 Test Program

The test program is written in the HP BASIC programming language in the HP BASIC operating system environment. HP BASIC is used because it provides a convenient environment for step by step control and debugging of the test program.

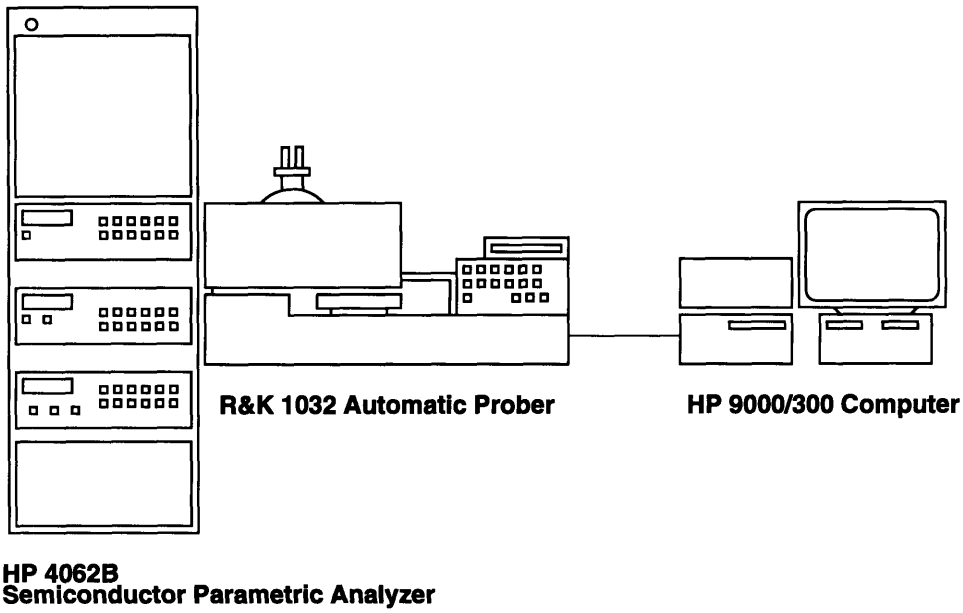

Figure 2-13: Equipment Setup

The test program controls the movement of the prober and the sequence of measurements. The subroutines for controlling the 4062B instruments are provided in HP's TIS program.

The DC four-point probe method is used on the Van der Pauw structures to obtain the sheet resistance of the lower conductor. Current is forced between two contacts at the opposite ends of resistive linewidth structure and the voltage drop across the line is measured from two contacts located between the current taps. The magnitude of the current is chosen so that the measured voltage is between 2.0 and 20mV [14]. The current is reversed and the two measured voltages are averaged in order to eliminate errors introduced by voltage offsets due to instrumentation errors. The linewidth is determined from measurements on the linewidth bridge structure in combination with measurements on the Van der Pauw structure. The sheet resistance is calculated using $R_S=(V/I)(\pi/\ln 2)$, and the linewidth is calculated using the formula $W=R_S$ * L* I/V [22].

The parasitic capacitance from the test setup is measured first at each test structure location by moving the chuck to the desired location and measuring the open

capacitance between the probe pin and the grounded chuck. This parasitic measurement is made for each of the capacitors in that module. The parasitic capacitance measured is later subtracted from the subsequent measured device capacitance.

After measuring the parasitic capacitance for each module, the chuck is raised until the probe pins come into contact with the probe pads on the wafer. The voltage is ramped from -3V to +3V with 1V steps on the top electrode and both the chuck and the bottom electrode are held at ground potential. The capacitance recorded is the average of seven capacitance measurements. An error checking routine is added to detect any large variation of capacitance during the voltage sweep.

A number of options are available within the test program. The user can specify testing the full wafer, select the die to be tested in any random order or in a certain step. The program also allows the user to specify the probe card dimensions so new probe cards with different pin configurations can be used.

The test program for the HP test wafers is written on HP's ICMS (Integrated Circuit Measurement System), a graphical user interface for defining semiconductor parametric tests. The reference parameters for the wafers and probe cards are defined in ICMS's Test Reference Area. The wafer information is organized into a natural hierarchy. The device, at the bottom of the hierarchy, is defined as an individual test structure. The next higher level is the module. The third level is the die which is a collection of modules. At the top of the hierarchy is the wafer.

For the Wafer Reference file, the wafer diameter, the die size, the die stepping distance, maximum number of rows and columns are entered. The user can also select only a subset of die by clicking on the wafer map. The Die Reference file contains the coordinates of each of the modules on the die and the module type which is defined in the module reference file. A Module Reference file has to be created for each of the modules. It contains the device type and terminal connection for each of the devices. The Device Reference file defines the terminal names for each of the devices.

After the Test Reference files are created, the Test Definition files have to be created. The Test Definition files have a hierarchy similar to that of the Test Reference. A cassette test consists of wafer tests associated with the individual wafers. A wafer

test consists of a selected set of die to be tested associated with a die test. A die test consists of a selection of modules and devices to be tested associated with one of more algorithms to measure the devices. The algorithms associated with each of the device types are created in the Algorithm Definition area and stored in an algorithm library.

2.7.3 Data Format / Data Collection

The data stored from the prober is a file with 5 columns: die number, subdie number, device number, and capacitance data or linewidth data. The sheet resistance data is also recorded for the Van der Pauw structures. The data is then sorted into 32 data files using a program that splits all the capacitance data and linewidth data into individual data files for each test structure.

The capacitance data for each of the data files is converted into thickness data using a presimulated capacitance-thickness data file. At this stage, the factor level for each of the test structures are appended into the individual data files for statistical analysis. The thickness data from each test structure is then merged into a single data file as an input to the S-plus statistical analysis program [2]. An analysis of variance (ANOVA) is then performed on the data file.

Chapter 3

Data Analysis and Results

The data analysis for both MIT and HP data are performed using the S-Plus statistical analysis program. Analysis methods include exploratory data visualization, analysis of variance, and Pareto plots, and simple spatial modeling and visualization.

3.1 MIT Results

The data obtained from each of the 12 wafers is individually analyzed. We will discuss the results from a typical wafer (B7) in this section.

3.1.1 Overall ILD Thickness Distributions

Figure 3-1 shows the histogram of ILD thickness for one of the MIT wafers (B7). The normal quantile-quantile plot in Figure 3-2 shows that the distribution deviates quite a bit from the normal distribution which is represented by the straight line. This implies that the observations of ILD thickness within this sample are not independent of one another, but that there is some systematic factor affecting the ILD thickness. Further analysis on this data indicate that this indeed is the case. The overall ILD thickness distribution for wafer B7 has a mean of 0.5724μm, a median of 0.562μm, and a standard deviation of 0.0532μm.

By separating the linewidth structures with 1.5μm linewidth from those with 5μm linewidth as shown in Figure 3-3, we can observe that the distributions are

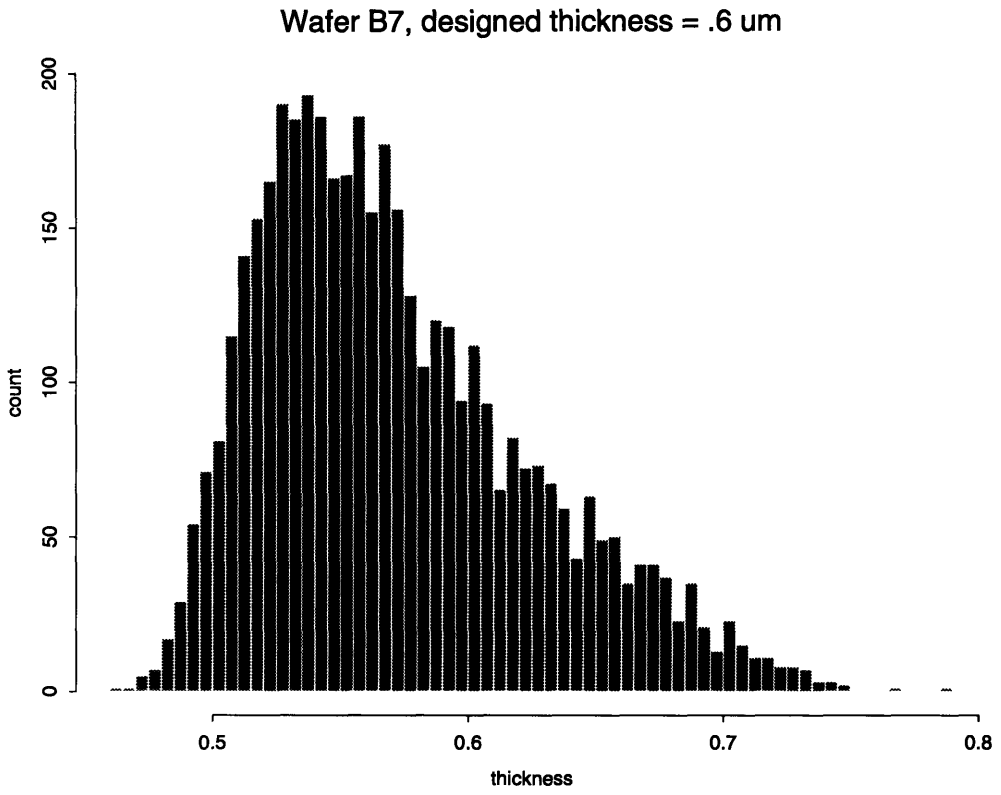

Figure 3-1: Histogram of ILD Thickness (Wafer B7)

shifted from one another which is an indication that linewidth is an important factor contributing to the ILD thickness. The median ILD thicknesses for structures with linewidth of 5μm is 0.572μm while those with linewidth of 1.5μm have a median of 0.553μm.

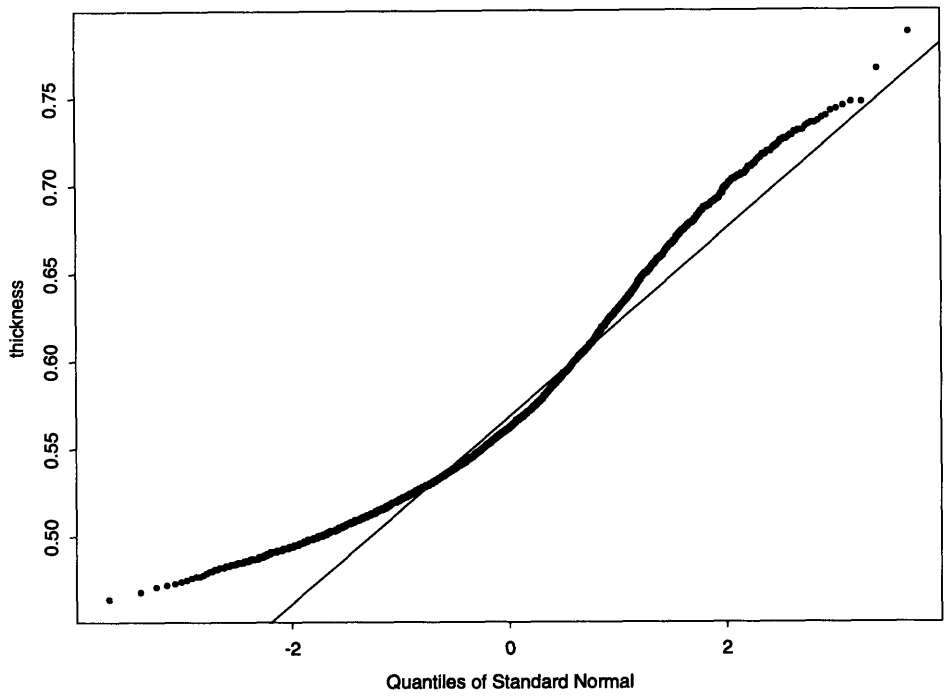

Figure 3-2: Normal Quantile-quantile Plot of ILD Thickness (Wafer B7)

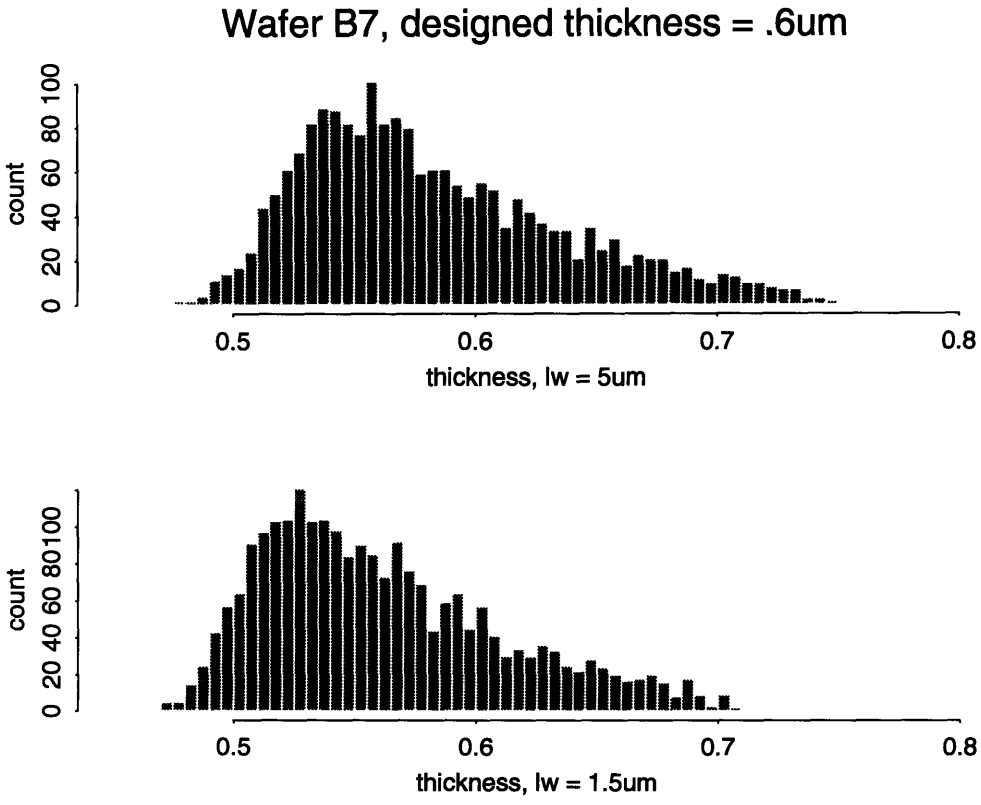

Figure 3-3: Comparison of ILD Thickness Distributions for the Linewidth Factor

3.1.2 Results from Analysis of Variance (ANOVA)

The Analysis of Variance (ANOVA) calculates the F value for each of the experimental factors and the interactions among them. The F value is the ratio of the mean square within each factor to the mean square among all factors. The sum of square within each factor, which is an estimate of within the group variance, is calculated from the difference between individual data and the mean for the factor. The sum of square among factors is calculated from the difference between individual factor means and the grand mean so it is an estimate of the inter-group variance. The sum of square divided by the degrees of freedom gives us the mean square value. Since the F value is the ratio of these two mean squares, a larger F means that the inter-group variance is large compared to the intra-group variance, and hence the result is more significant [17]. The residual is the fraction of the result not explained by any of the factors and their interactions. Further investigation is needed to clarify the allocation of residual degrees of freedom in the face of systematic spatial variation. In particular, "split plot" considerations should be investigated [24].

Table 3.1 summarizes the Analysis of Variance results for wafer B7. From the table, we can see both linewidth and line spacing are highly significant.

Figure 3-4 shows the relative contributions of each factor to the ILD thickness. The largest contributor here is the y-coordinate which accounts for about 80 percent of the variation. The linewidth and line spacing also have significant effect on the ILD thickness.

3.1.3 Spatial Dependence of ILD Thickness

The large scale spatial dependence can also be observed in Figure 3-5 where we have plotted the ILD thickness against the die number. The plot on the left side represents the structures with designed linewidths of 5μm and the one on the right represents those with designed linewidths of 1.5μm. Each point in the figure represents an ILD thickness data point over a test structure. The ILD thickness seems to follow a periodic wiggled pattern. The periodicity comes from the fact that the die are split

	Df	Sum of Sq	Mean Sq	F Value	Pr(F)
lw	1	0.49305	0.493045	192.3897	0
ls	1	0.22137	0.2213745	86.3819	0
fl	1	0.0269	0.0269019	10.4973	0.0012046
nof	1	0.11018	0.1101788	42.9926	0
go	1	0.03159	0.0315918	12.3273	0.000451
id	1	0.04892	0.0489151	19.087	1.28e-05
lw:ls	1	0.05856	0.0585561	22.849	1.8e-06
lw:fl	1	0.00034	0.0003379	0.1318	0.7165483
ls:fl	1	0.0114	0.0114005	4.4485	0.0349887
lw:nof	1	0.00184	0.0018382	0.7173	0.3970861
ls:nof	1	0.03225	0.0322467	12.5829	0.0003934
fl:nof	1	0.00558	0.0055766	2.176	0.1402502
lw:go	1	0.0162	0.0162041	6.323	0.0119545
ls:go	1	0.00408	0.0040781	1.5913	0.2072062
fl:go	1	0.00884	0.0088376	3.4485	0.0633781
nof:go	1	0.00239	0.002391	0.933	0.3341429
lw:id	1	0.01244	0.0124356	4.8524	0.0276597
ls:id	1	0.00087	0.0008703	0.3396	0.5600914
fl:id	1	0.04837	0.0483736	18.8757	1.43e-05
nof:id	1	0.00112	0.0011174	0.436	0.5090772
lw:ls:fl	1	0.00952	0.0095245	3.7165	0.0539416
lw:ls:nof	1	0.01258	0.0125812	4.9093	0.0267648
lw:fl:nof	1	0.00053	0.0005332	0.208	0.6483287
lw:ls:go	1	0.01884	0.0188439	7.353	0.0067216
lw:fl:go	1	0.02155	0.0215488	8.4085	0.0037536
lw:nof:go	1	0.00936	0.0093624	3.6533	0.0560253
ls:nof:go	1	0.02706	0.0270555	10.5572	0.0011662
Residuals	4305	11.0326	0.0025627		

Table 3.1: ANOVA Results for Wafer B7

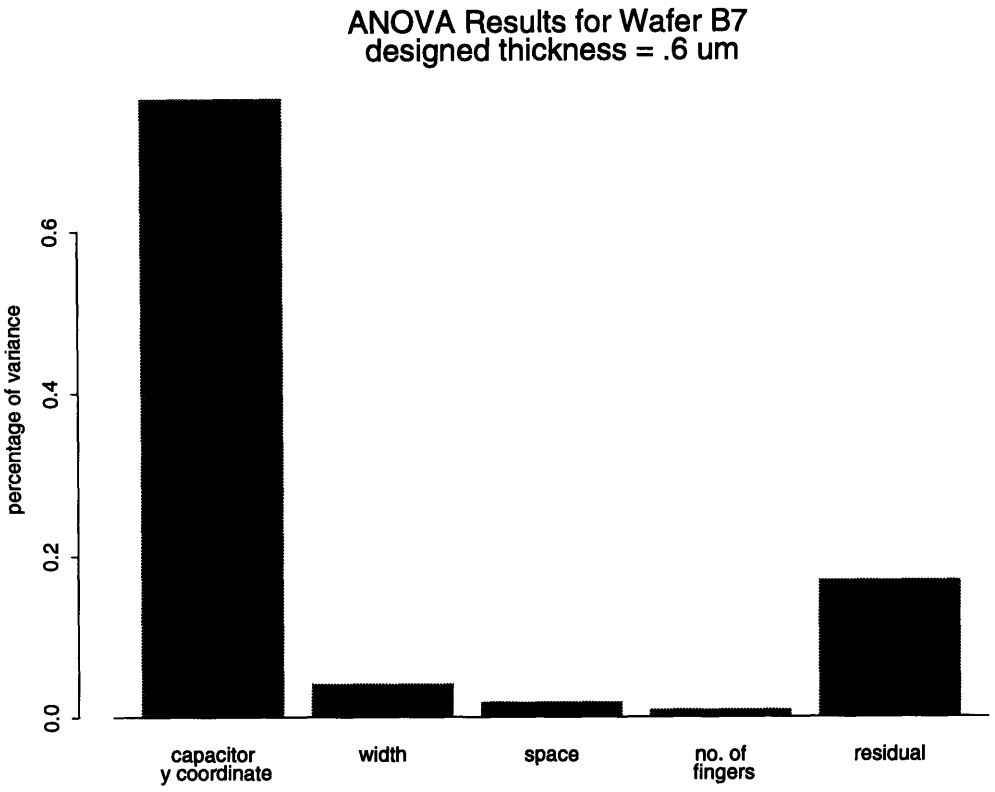

Figure 3-4: Relative Contributions of each Factor on ILD thickness

into seven rows, and the die are numbered in a horizontal snake pattern, with rows from the top to the bottom of the wafer. Closer examination of the Figure 3-5 reveals that the number of period humps in the ILD thickness data is equal to the number of rows of die. The ILD thickness decreases towards the bottom of the wafer where the die have higher numbers.

There are two possible explanations for this phenomenon. First of all, the chemical reaction in the BPSG deposition step could be affected by furnace temperature or gas flow nonuniformity, resulting in a thicker dielectric at the bottom of the wafers. However, this explanation is ruled out because the wafer orientations are randomized during this process even though we observe similar spatial dependence in all our wafers. The other explanation relates to the orientation of the wafers during the BPSG reflow process. Since the wafers are all inserted with their flats facing down, the dielectric would naturally flow down towards the flats which makes the dielectric closer to the flats thicker.

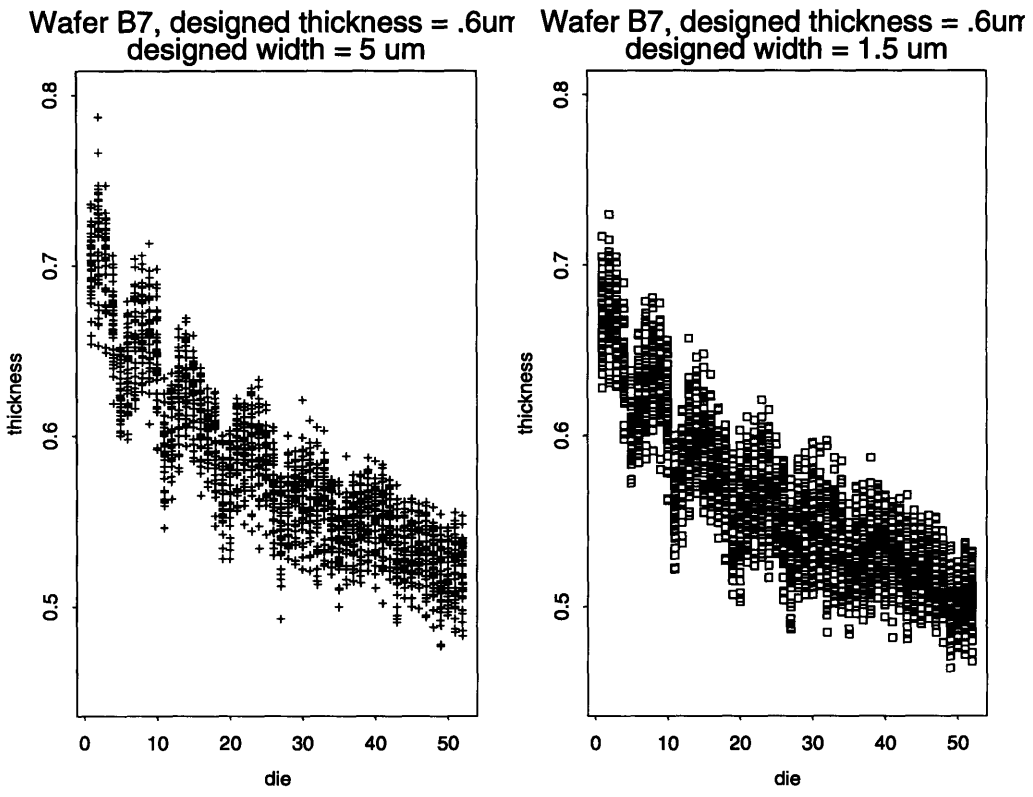

Figure 3-5: Scattered Plots of ILD thickness (Wafer B7)

A three dimensional perspective view of the ILD thickness for wafer B7 is shown in Figure 3-6. The ILD thickness is interpolated between adjacent test structures.

3.2 HP Results

A total of eleven wafers are tested for the HP experiment. We look at the data from wafer M2 in this section. All ILD thicknesses have been arbitrarily and consistently normalized to disguise absolute HP process ILD thicknesses.

3.2.1 Overall Distribution of ILD Thickness

The ILD thicknesses for the HP wafers are quite normally distributed as shown in Figure 3-7. This is verified by the normal quantile-quantile plot in Figure 3-8. The distribution diverges from the straight line representing the normal distribution at the ends because there are a number of data points which lie below 0.3μm and above

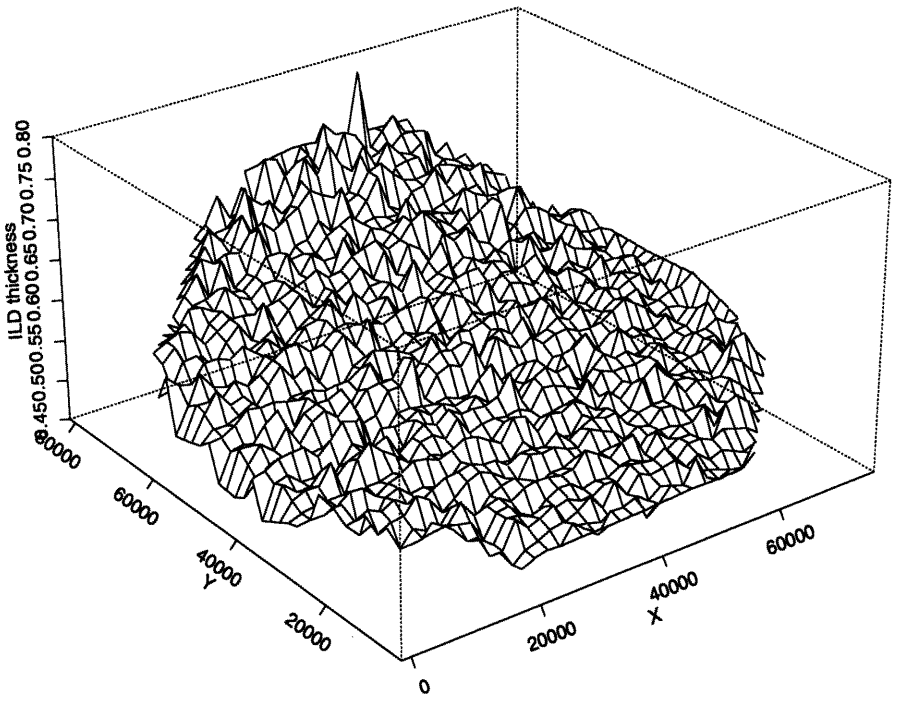

Figure 3-6: Three Dimensional Perspective View of ILD thickness (Wafer B7)

0.9μm. These outliers correspond consistently to a few test structures. However, the reason for these structures having thicknesses that differ so much from the majority of the structures is still not well understood.

3.2.2 Spatial Dependence of ILD Thickness

The ILD thicknesses for the HP wafers also display significant spatial dependencies. As in the MIT wafers, the y coordinate dependency as seen in Figure 3-10 is much stronger than the x coordinate dependency shown in Figure 3-11.

In Figure 3-12, we have plotted the ILD thickness against the radial distance from the center of the wafer. It shows that the structures further away from the wafer center have larger ILD thicknesses. The 3-D perspective plot of ILD thickness in Figure 3-14 also shows this pattern where the edge ILD thickness is larger than the middle. Individual test structures also follow this collective pattern. Figure 3-13 shows the ILD thickneses for test structure 3. The pattern displays a close resemblance to

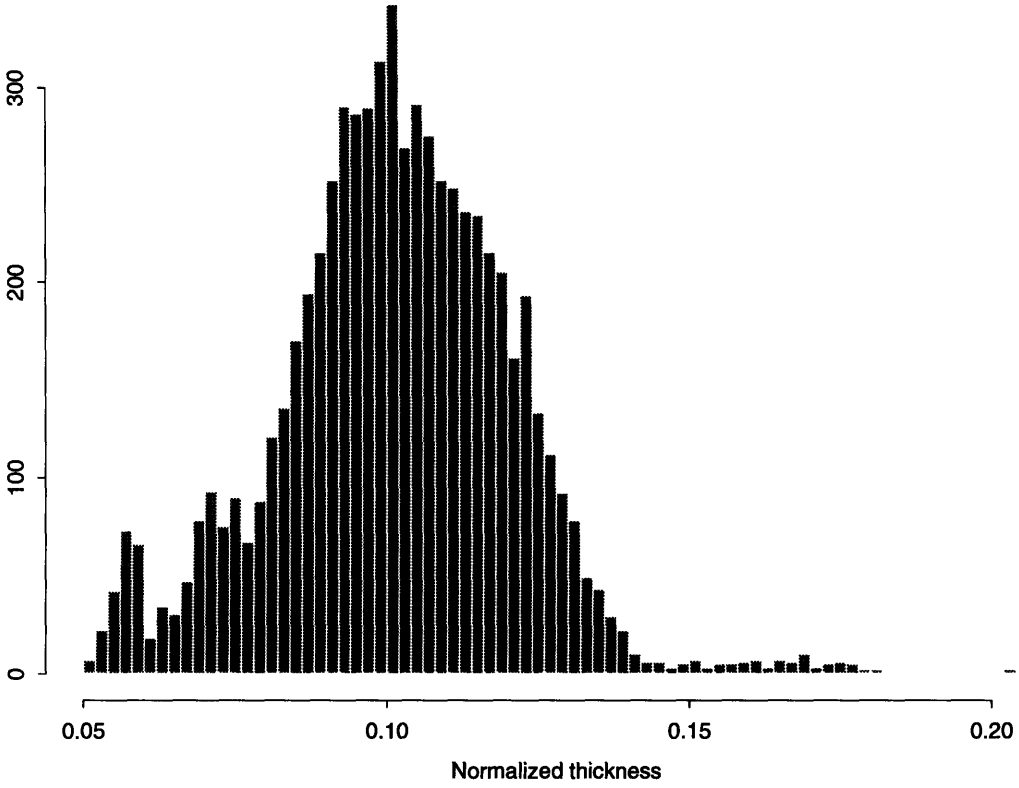

Figure 3-7: Histogram of ILD Thickness (Wafer M2)

Figure 3-12. Individual analysis on the other test structures shows similar results.

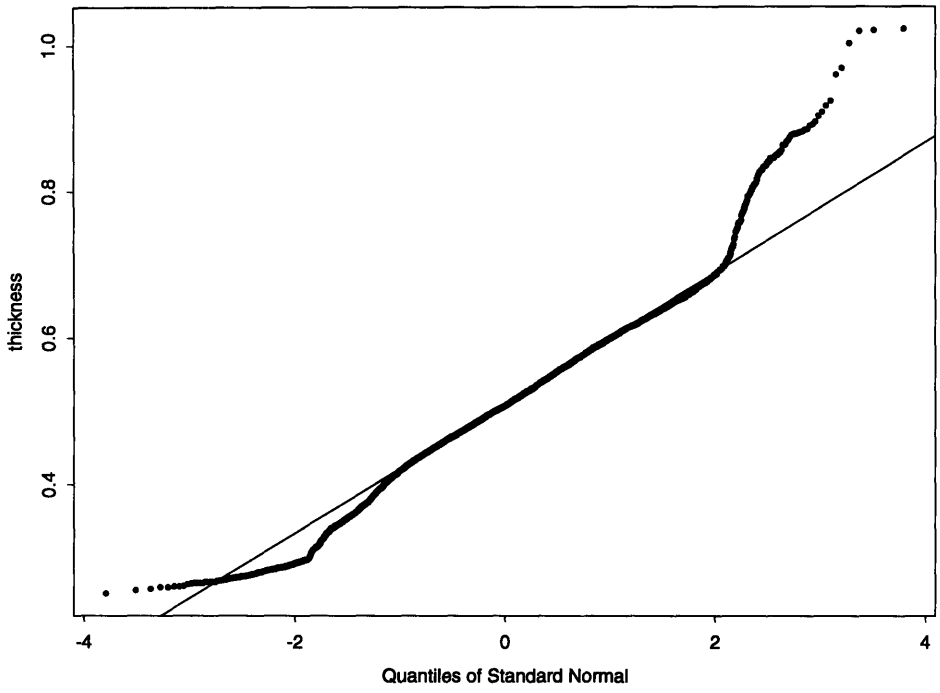

Figure 3-8: Normal Quantile-Quantile Plot (Wafer M2)

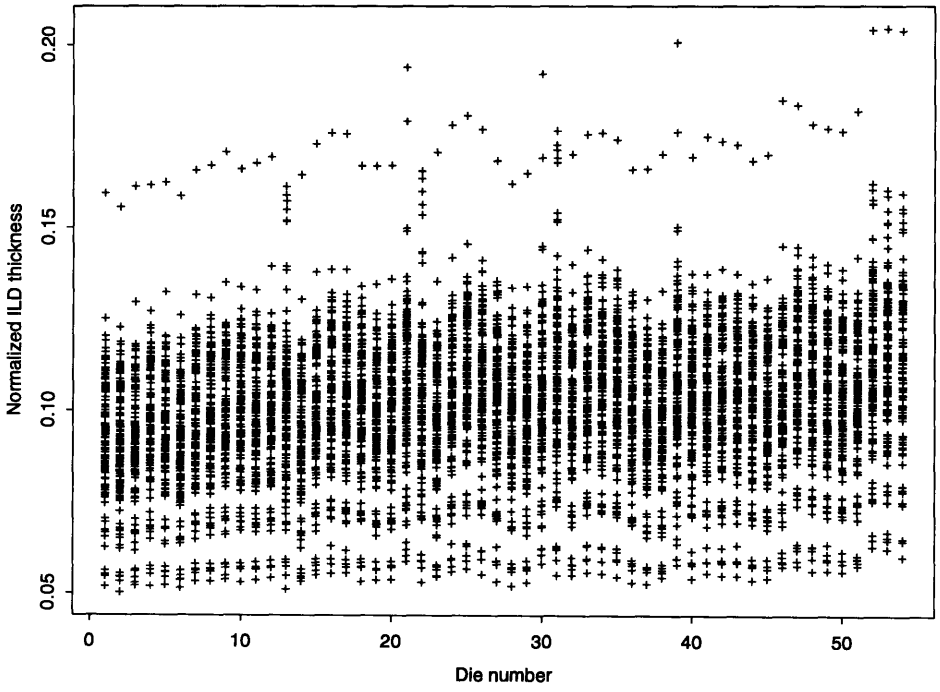

Figure 3-9: Scattered Plots of ILD Thickness (Thickness vs Die Number)

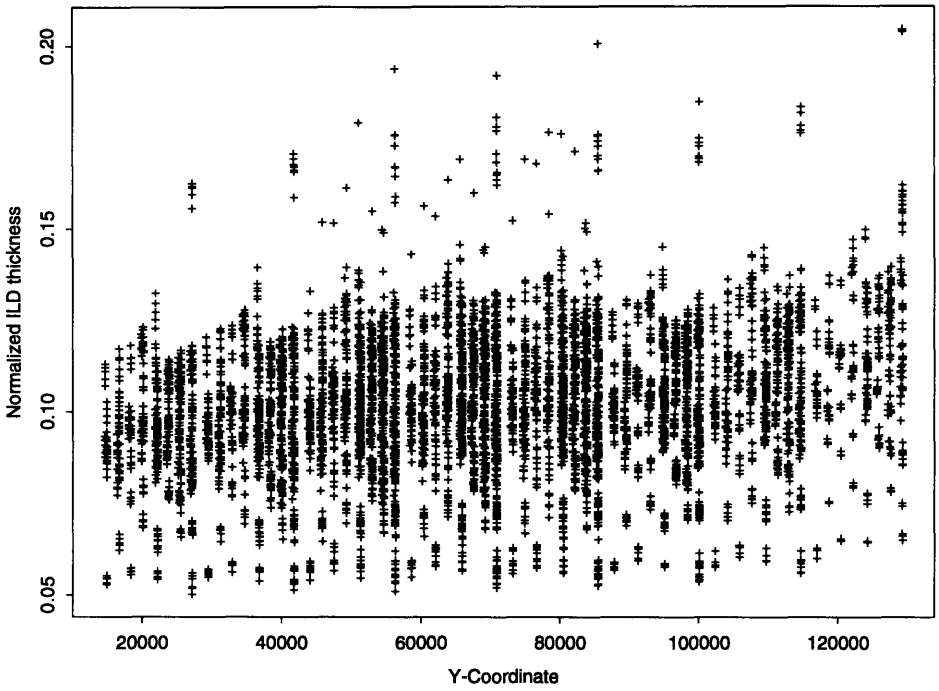

Figure 3-10: Scatter Plots of ILD Thickness (Thickness vs. Y-Coordinate)

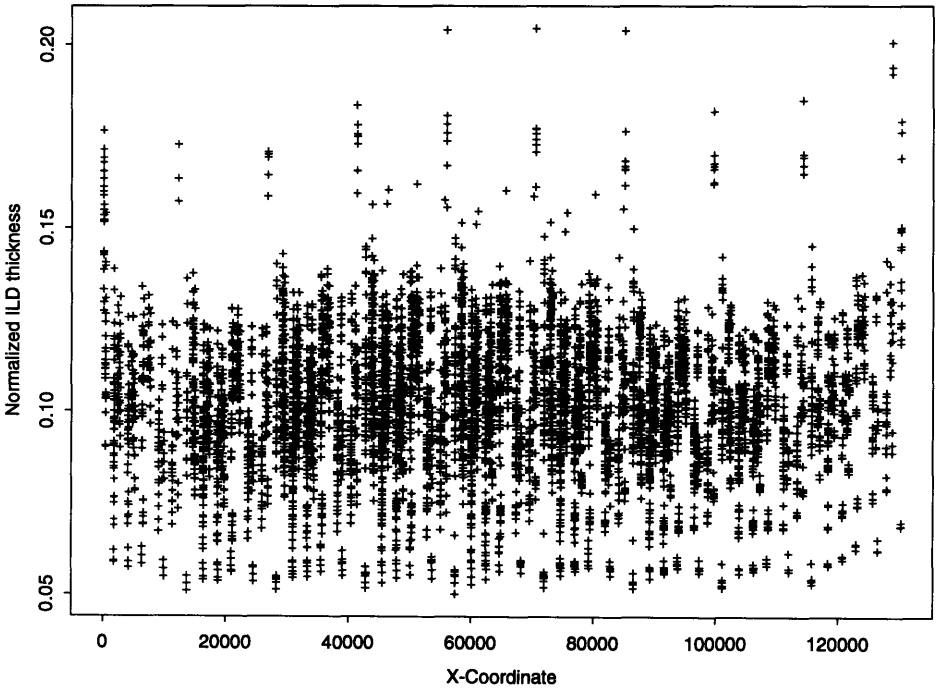

Figure 3-11: Scatter Plots of ILD Thickness (Thickness vs. X-Coordinate)

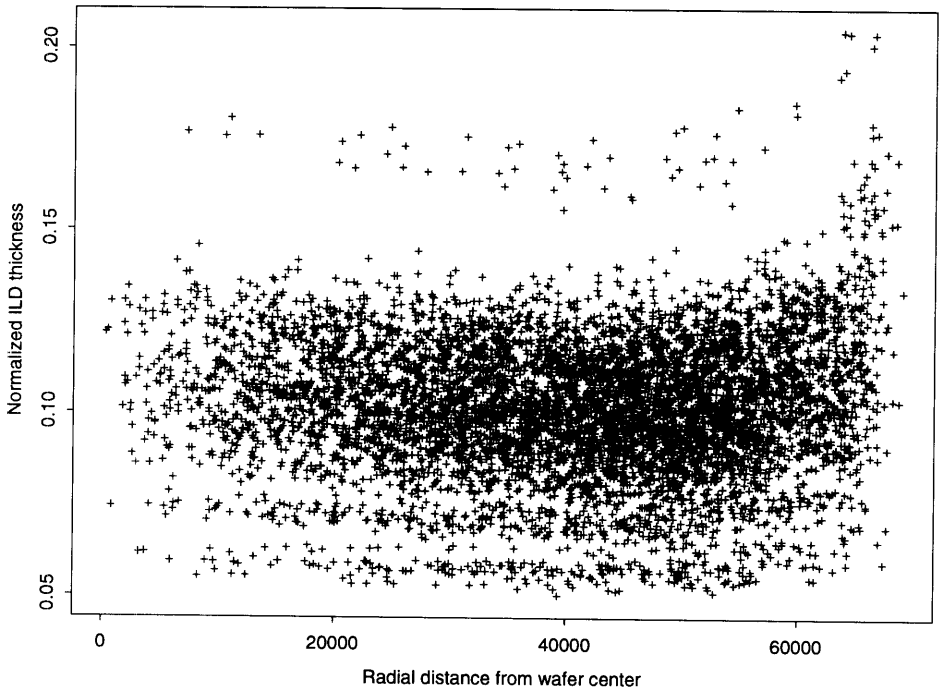

Figure 3-12: Scatter Plots of ILD Thickness (Thickness vs. Distance from Center)

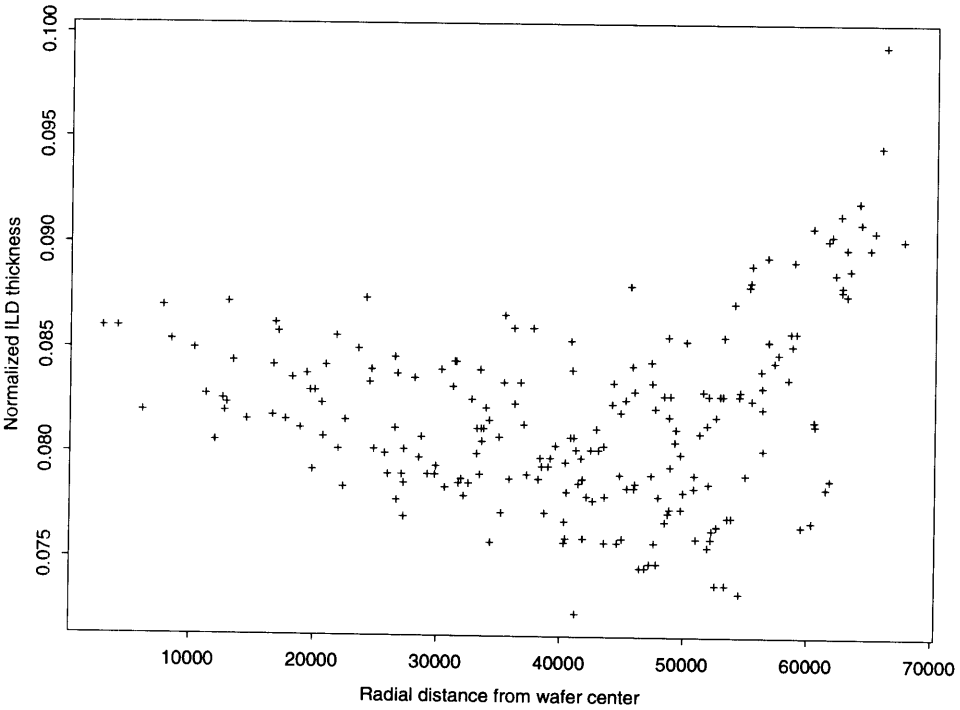

Figure 3-13: ILD Thickness vs. Radial Distance from Wafer Center (Structure 3)

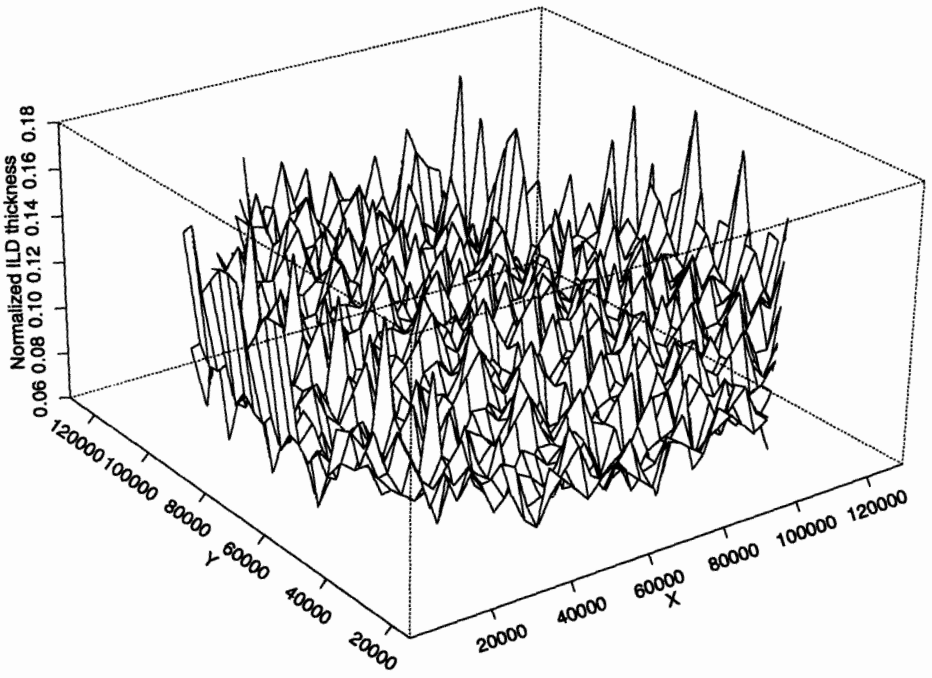

Figure 3-14: Three Dimensional Perspective View of ILD thickness (Wafer M2)

3.2.3 ANOVA Analysis Results and the Layout Factor Dependence

Looking at the main effects plot in Figure 3-15, it appears that linewidth is the largest factor influencing the ILD thickness in the CMP process.

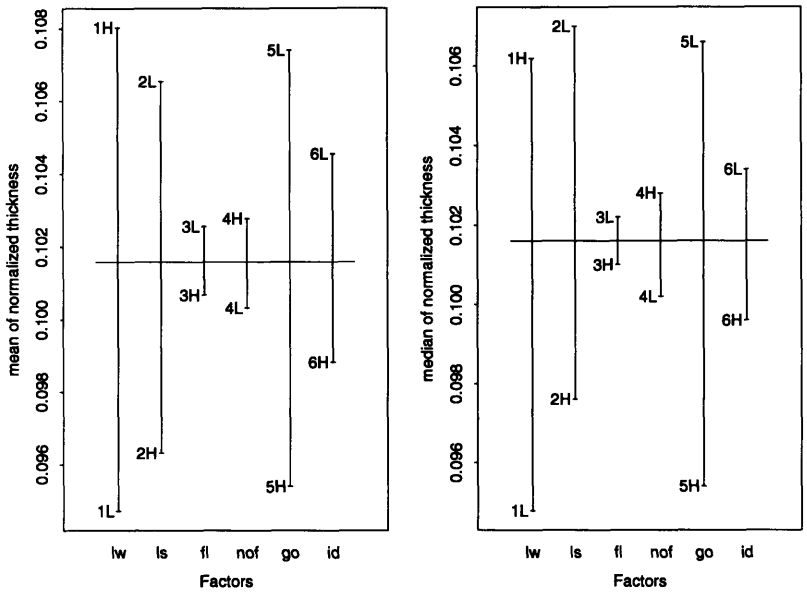

Figure 3-15: Main Effects Plot (Wafer M2)

More detailed examination of the results of ANOVA in Table 3.2, however, indicates that the second-level interaction between the line length and the number of lines is the dominant factor. This makes intuitive sense because we expect CMP to be more area dependent and the line length and number of lines are both directly proportional to the area. The main effect plot also confirms our intuition regarding the linewidth and line spacing's effect on ILD thickness. The structures with larger linewidths have larger ILD thickness while those with smaller spacing have larger ILD thickness. Also, the structures with larger number of lines have larger ILD thickness. The geometric orientation of the capacitor seems to have a significant effect due to the fact that the wafers rotate in a single direction on the CMP machine. The effect of the interaction ring on the ILD thickness, however, is counter-intuitive because the structures with interaction rings appear to have smaller ILD thickness. This can only be explained by more detailed analysis of the physics behind chemical-mechanical

polishing. Overall, the CMP process seems to have a large number of significant factors affecting the ILD thickness, and the second and third level interactions seem to play a larger role.

3.2.4 Analysis Results for Area Intensive Structures

Before testing the HP wafers on the automatic prober, manual measurements on the area intensive capacitors were made assuming the metal lines have linewidths as drawn. The results from these measurements are shown in Figure 3-16. The measurements seem to indicate a large area dependency of ILD thickness. The dependence on the interaction rings seems to be relatively small. Our intent is to compare this data with the automatically measured data.

ILD Thickness Distribution

Figure 3-17 through Figure 3-19 show the distribution of ILD thickness for all the area intensive structures. From the histogram in Figure 3-17, the ILD thickness over these structures appears to be approximately normally distributed.

	Df	Sum.of.Sq	Mean.Sq	F.Value	Pr.F.
fl:nof	1	9.97444	9.974437	3710.841	0
lw	1	7.39765	7.397647	2752.185	0
go	1	6.99315	6.993148	2601.697	0
ls:go	1	5.85383	5.853828	2177.83	0
ls	1	4.7614	4.761396	1771.407	0
ls:fl:go	1	1.82621	1.826211	679.415	0
lw:go	1	1.81762	1.817624	676.22	0
fl:id	1	1.37718	1.377175	512.358	0
lw:ls:go	1	0.92318	0.923176	343.454	0
ls:nof:go	1	0.85105	0.851047	316.619	0
nof:go	1	0.81543	0.815433	303.37	0
id	1	0.79625	0.796251	296.233	0
ls:nof	1	0.61411	0.614112	228.471	0
nof	1	0.43845	0.438447	163.118	0
lw:nof:go	1	0.33726	0.337261	125.473	0
go:id	1	0.25401	0.254011	94.501	0
lw:ls	1	0.22633	0.226327	84.202	0
lw:fl:go	1	0.21892	0.218919	81.446	0
lw:fl	1	0.16291	0.162913	60.609	0
nof:id	1	0.15538	0.155381	57.807	0
ls:id	1	0.11638	0.116381	43.298	0
ls:fl	1	0.06622	0.066224	24.638	7e-07
fl	1	0.04857	0.048566	18.068	2.16e-05
lw:fl:nof	1	0.03694	0.036939	13.743	0.0002113
lw:ls:nof	1	0.01476	0.014757	5.49	0.0191554
lw:nof	1	0.01183	0.011826	4.4	0.0359825
ls:fl:nof	1	0.00913	0.009129	3.396	0.0653844
fl:go	1	0.00037	0.000372	0.138	0.7098717
lw:ls:fl	1	9e-05	8.8e-05	0.033	0.8563485
lw:id	1	8e-05	7.9e-05	0.029	0.8642058
Residuals	6664	17.91228	0.002688		

Table 3.2: Summary of ANOVA Results for Wafer M2

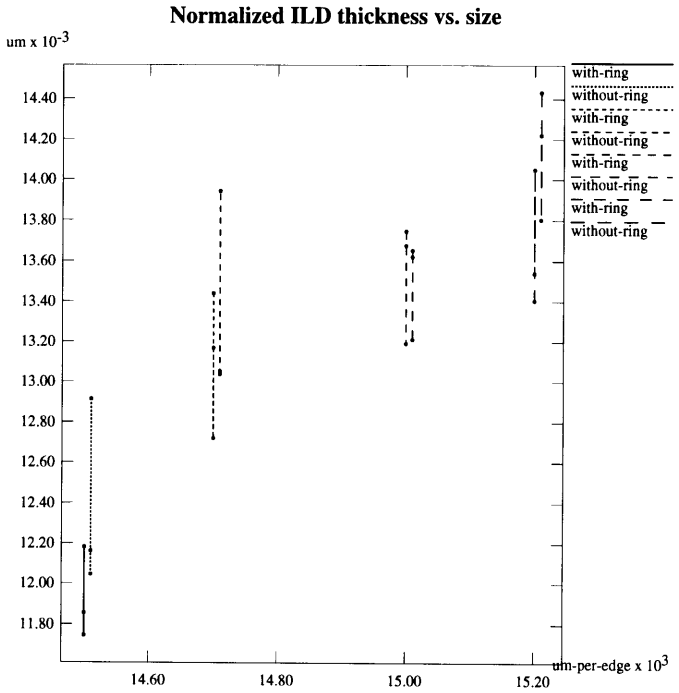

Figure 3-16: Manually Measured ILD Thickness Data for Various Capacitor Sizes

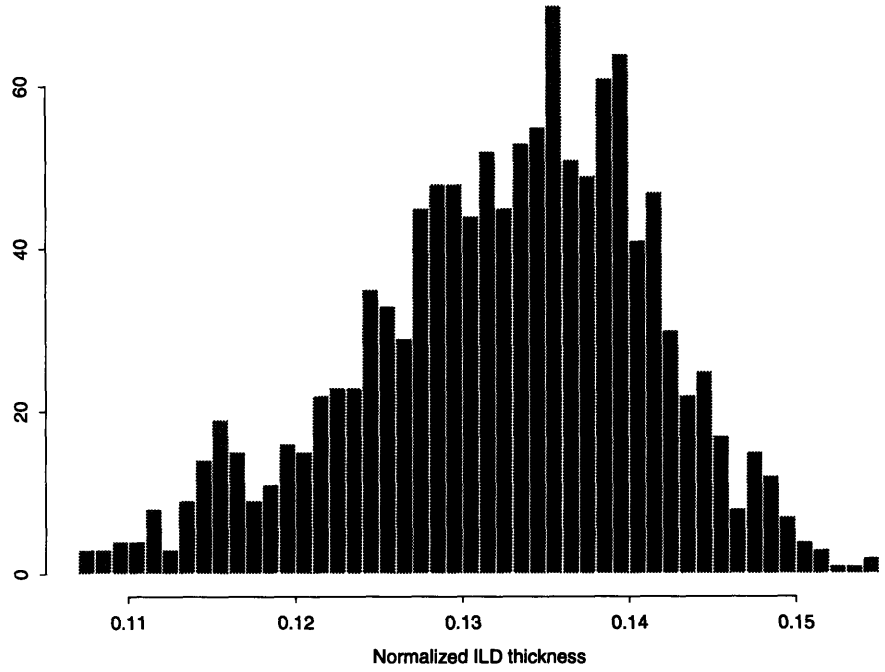

Figure 3-17: Histogram of ILD Thickness for Area Intensive Structues

ANOVA Results for Area Intensive Structures

Table 3.3 summarizes the results for the ANOVA of area intensive structures. The significant factors here are the y coordinate, interaction between the x and y coordinates, the x coordinate, the interaction ring, and the size of the structure. It is surprising that the area has a smaller effect than the interaction ring. Perhaps adding various ring sizes and interaction distances into our experimental design could give us a better understanding of the proximity effect.

The factor plot in Figure 3-20 gives us the information on the variability, skewness, and outliers in the response for each level of each experimental factor. The bar in the middle of the box encodes the median of the distribution. The upper and lower ends of the box are upper and lower quartiles. The dashed lines encode the adjacent values. The upper adjacent value is the largest observation that is less than or equal to the upper quartile plus 1.5 times the interquartile range. The lower adjacent

	Df	Sum.of.Sq	Mean.Sq	F.Value	Pr.F.
yc	1	0.696655	0.6966548	550.7254	0
xc:yc	1	0.157513	0.1575127	124.5183	0
xc	1	0.130713	0.1307135	103.3327	0
ring	1	0.086583	0.0865831	68.4464	0
size	3	0.051798	0.0172659	13.6492	0
size:ring	3	0.037601	0.0125337	9.9082	1.9e-06
size:xc	3	0.007552	0.0025173	1.99	0.1136616
size:ring:xc	3	0.002899	0.0009662	0.7638	0.5143721
ring:xc	1	0.000699	0.0006987	0.5523	0.4575139
size:yc	3	0.001038	0.000346	0.2735	0.8445385
size:ring:yc	3	0.000815	0.0002717	0.2148	0.8861745
size:xc:yc	3	0.000548	0.0001828	0.1445	0.9332325
size:ring:xc:yc	3	0.00025	8.33e-05	0.0659	0.9779688
ring:yc	1	5.4e-05	5.41e-05	0.0428	0.8361587
ring:xc:yc	1	6e-06	5.8e-06	0.0046	0.945897
Residuals	1264	1.598931	0.001265		

Table 3.3: Summary of ANOVA Results for Area Intensive Structures

value is defined similarly [21]. From this box plot, we can observe that both the capacitor sizes and the interaction rings have rather constant variability and very little skew. The plot also shows that the ILD thickness is not monotonically dependent on the capacitor size. However, due to the global nature of the CMP process, there may be larger wafer-scale pattern dependencies that can only be captured by test structures much larger than ours. Further experimentation with the CMP process should certainly take this into consideration.

The data from the automated measurements seem to disagree with that from the manual measurements. This discrepancy is probably due to the fact that a random sample of test structures do not accurately represent the profile of ILD thickness. The lack of measurements for the metal linewidth variations could also account for this. Further independent analysis using SEM cross-sections is needed to help resolve these inconsistencies.

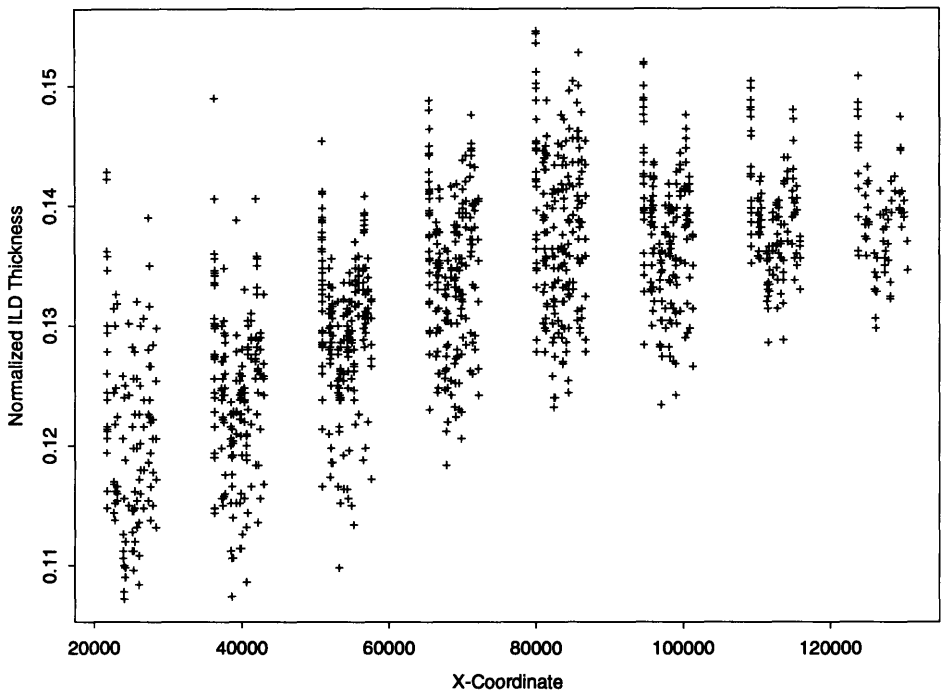

Figure 3-18: ILD Thickness vs. X-Coordinate Plotted for Area Intensive Structures

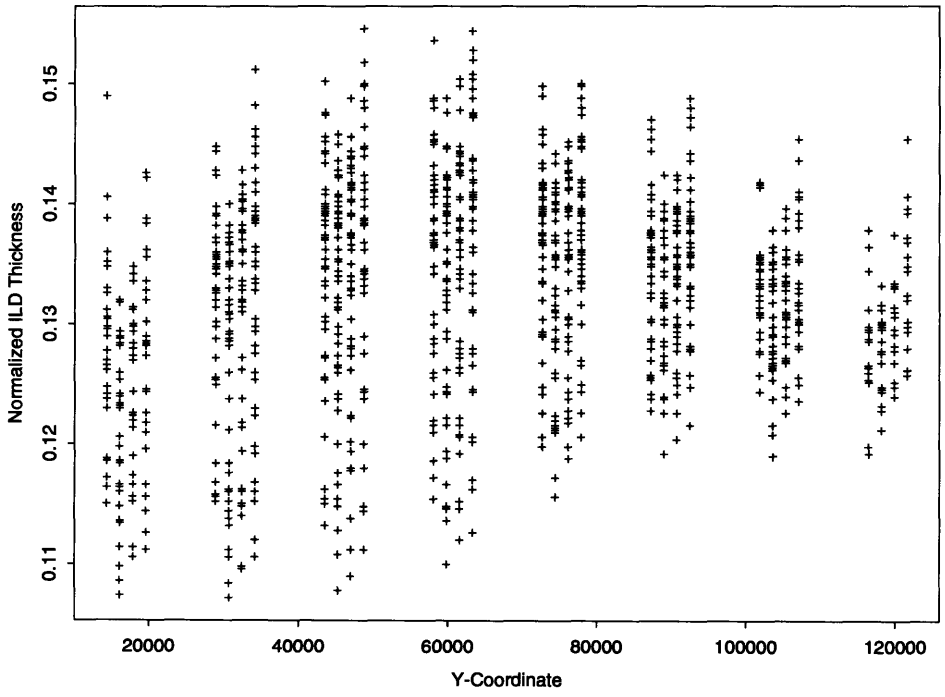

Figure 3-19: ILD Thickness vs. Y-Coordinate Plotted for Area Intensive Structures

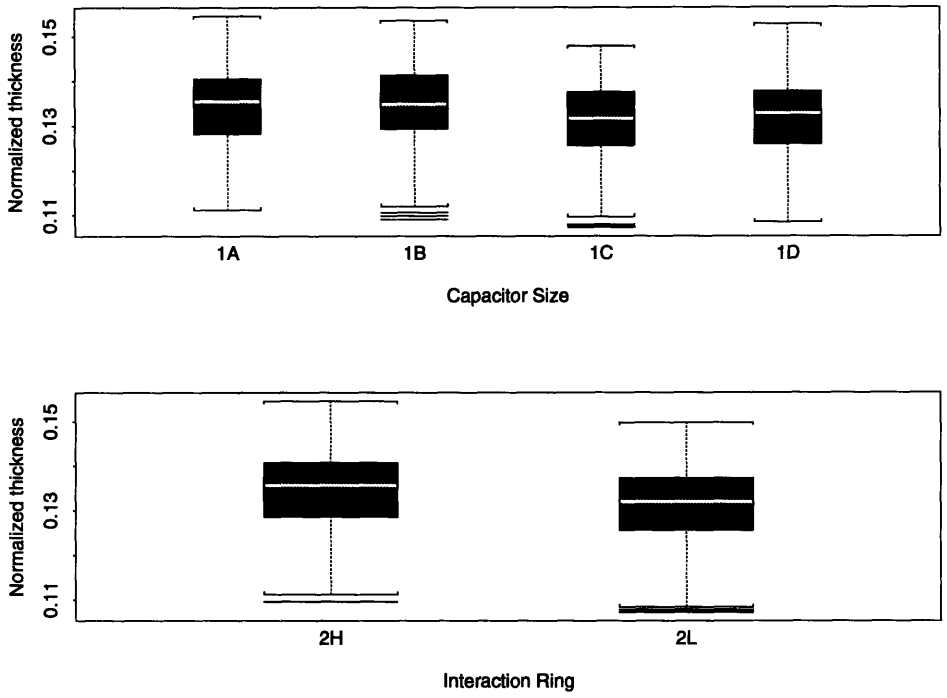

Figure 3-20: Factor Plot for Area Intensive Structures

Chapter 4

Conclusion

4.1 Summary of Results

Improving the yield levels in semiconductor manufacturing involves reducing the variabilities associated with each processing step. In order to reduce variabilities, we need to have means to identify and quantify these variabilities and determine their sources. Statistical metrology serves as a way to meet these goals by improving data collection and analysis methods. The results of statistical metrology could be applied in process development, equipment selection, and design for manufacturability.

In this thesis, we have demonstrated a working methodology for statistical metrology of interlevel dielectric thickness. This methodology involves design of electrical test structures using statistical design of experiments, using carefully designed short-loop flow processes, coupling with technology CAD (TCAD) tools, and data analysis using statistical techniques. We have applied the methodologies for MIT's planarization process using reflowed BPSG, and HP's process using TEOS planarized by chemical-mechanical polishing, and identified the factors that have the most significant impact on the ILD thickness. Spatial dependencies which are related to particular equipment and process recipes are also identified. We have seen the importance of identifying all possible sources of variation in statistical metrology. As we move towards applying the methodology to more sophisticated technology and longer process flows, this task will become more difficult and careful experimental design

will be needed.

4.2 Future Work

There are a number of investigations which could help us better understand the nature of ILD thickness variations. The spatial dependencies of ILD thickness could be decoupled from the layout dependencies by applying statistical filtering techniques with fast Fourier transforms (FFT) as suggested in [6]. Other forms of explicit spatial modeling should also be pursued [23]. Further SEM analysis is needed on the CMP wafers to verify the correspondence between the electrically measured ILD thickness and the physical thickness.

Another issue which warrants investigation is improvement of the measurement methodology. The parasitic capacitance between the probe pads and the chuck seems to be quite large. We could reduce this by either reducing the pad size or shielding the probe pins. We could also reduce the parasitic from the process aspect by making the field oxide thicker or by using higher level metals which would both reduce the effective distance between the pads and the ground plane.

Plans are already in progress to redesign our first generation mask. This second generation mask will include more levels of each of the important factors. From that second round experiment, we should be able to build a finer predictive model for ILD variation as a function of the key parameters. Large resistors have also been added next to the capacitors to determine size dependency of the polysilicon linewidth. The process used for this experiment will be modified for a more planar dielectric with either reflowed BPSG or an alternative dielectic planarized by chemical mechanical polishing. We also hope to be able to apply the methodology to relate the impact of ILD thickness variation on circuit performance by integrating simple circuits such as ring oscillators with ILD thickness test structures.

The continued development of a statistical metrology methodology for the identification and quantification of variation sources can help meet future requirements for improved process control and circuit performance.

Bibliography

[1] George E. P. Box, William G. Hunter, and J. Stuart Hunter, *Statistics for Experimenters: An Introduction to Design, Data Analysis, and model Building*, John Wiley & Sons, New York, 1978.

[2] Statistical Sciences, *S-PLUS User's Manual, Version 3.2*, Seattle: StatSci, a division of MathSoft, Inc., 1993.

[3] H.-Y. Kim, "Design of a test mask for the statistical metrology of VLSI interlevel dielectrics," Bachelor's Thesis, Electrical Engineering and Computer Science Department, MIT, May 1994.

[4] Raphael Manual. Technology Modeling Associates, Inc. Palo Alto, 1994.

[5] K.-J. Chang, S.-Y. Oh, N. Chang, L. Mui, S. Peng, K. Young, and P. Raje, "Nondestructive multilevel interconnect parameter characterization for high-performance manufacturable VLSI technologies," *Digest of Technical Papers of the IEEE 1993 Symposium on VLSI Technology,* pp.136-136. Kyoto, Japan, May 1993.

[6] C. Yu, T. Maung, C. Spanos, D. Boning, J. Chung, H.-Y Liu, K.-J. Chang, and D. Bartelink, "Use of short-loop electrical measurements for yield improvement," to appear in the *IEEE Transactions on Semiconductor Manufacturing,* 1995.

[7] Duane Boning, Tinaung Maung, James Chung, Keh-Jeng Chang, Soo-Young Oh, and Dirk Bartelink, "Statistical metrology of interlevel dielectric thickness variation," *Proceedings of the SPIE Symposium on Microelectronic Manufacturing*, SPIE Vol. 2334, October, 1994.

[8] Victor Comello, "Planarizing Leading Edge Devices," *Semiconductor International*, November, 1990.

[9] Peter Singer, "Searching for Perfect Planarity," *Semiconductor International*, March, 1992.

[10] Stanley Wolf, *Silicon Processing for the VLSI Era, Volume 2*, Lattice Press, 1993.

[11] *ICMS User's Manual*, Hewlett-Packard Corporation, 1991.

[12] *HP 4062B Manual*, Hewlett-Packard Corporation, 1989.

[13] P. K. Tedrow and C. G. Sodini, "MIT Twin Well CMOS Process, Version 1.3," MIT Microsystems Technology Laboratories internal circulation, July, 1993.

[14] William B. Glendinning and John N. Helbert, editors, "Handbook of VLSI microlithography : principles, technology, and applications," p.216, Noyes Publications, Park Ridge, N.J., 1991.

[15] G. P. Carver, L. W. Linholm, and T. J. Russell, "Use of microelectronic test structures to characterize IC materials, processes, and processing equipment," *Solid State Technology*, September, 1980.

[16] M. G. Buehler, S. D. Grant, and W. R. Thurber, "Bridge and Van der Pauw sheet resistors for characterizing the linewidth of conducting layers," *Journal of Electrochemical Society*, p. 650, April, 1978.

[17] Geoffrey Norman and David Streiner, "PDQ Statistics," B.C. Decker, Inc., Toronto, 1986.

[18] Srinivasan Sivaram, Hubert Bath, Robert Leggett, Alvaro Maury, Kenneth Monning, and Tobert Tolles, "Planarizing interlevel dielectrics by chemical mechanical polishing," *Solid State Technology*, p. 87, May, 1992.

[19] Iqbal Ali, Sudipto Roy, and Greg Shinn, "Chemical-mechanical polishing of interlayer dielectric: A review," *Solid State Technology*, p. 63, October, 1994.

[20] Peter Johnson, and Prochy Sethna, "Using BPSG as an Interlayer Dielectric," *Semiconductor International*, p. 80, October, 1987.

[21] William S. Cleveland, *Visualizing Data*, AT&T Bell Laboratories, Murray Hill, New Jersey, 1993.

[22] Dieter K. Schroder, *Semiconductor material and device characterization*, Wiley, New York, 1990.

[23] John K. Kibarian and Andrzej J. Strojwas, "Using Spatial Information to Analyze Correlations Between Test Structure Data," *IEEE Transactions on Semiconductor Manufacturing*, p. 219, August, 1991.

[24] D. Doherty Fitzgerald, "Analysis of polysilicon critical dimension variation for submicron CMOS processes," Master's Thesis, MIT, June 1994.

[25] Soo-Young Oh and Keh-Jeng Chang, "2001 Needs for Multi-Level Interconnect Technology," *IEEE Circuits & Devices*, January, 1995.

Appendix A

HP Mask Modules

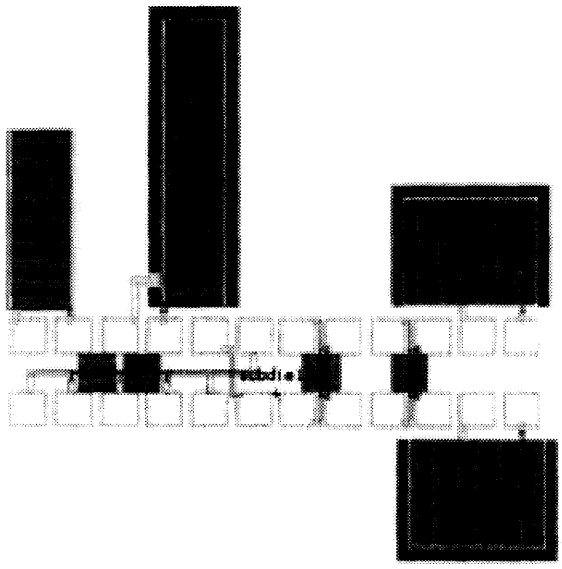

Figure A-1: Module 1

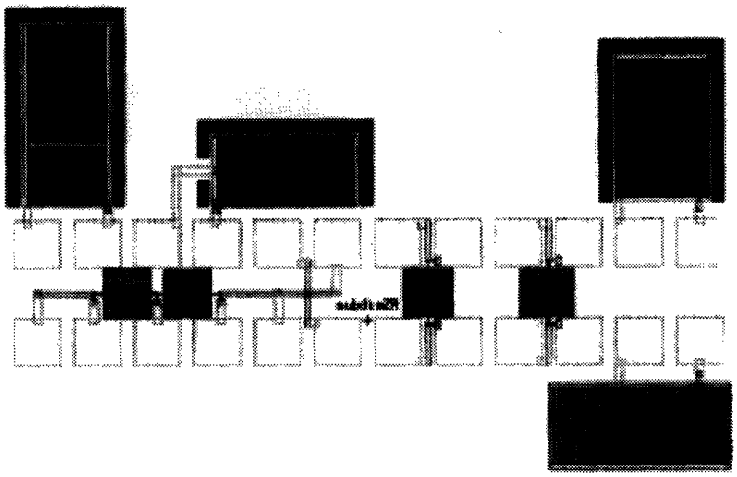

Figure A-2: Module 2

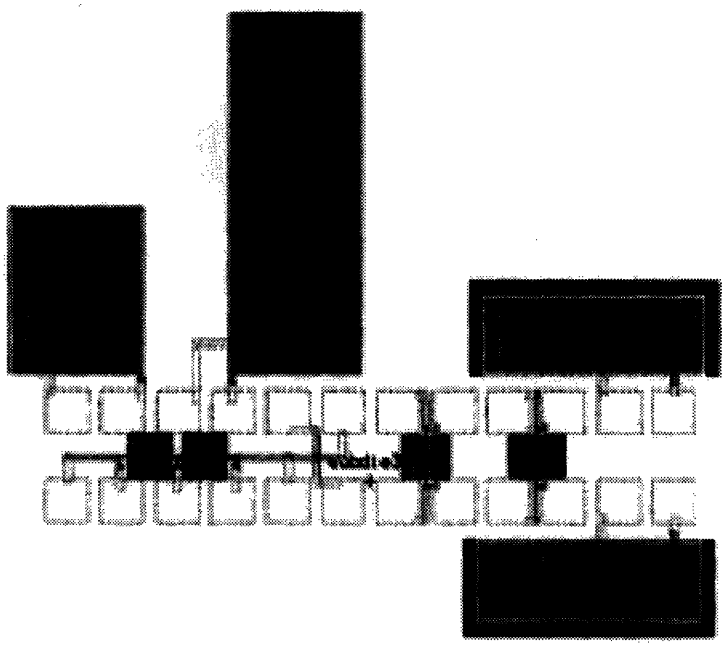

Figure A-3: Module 3

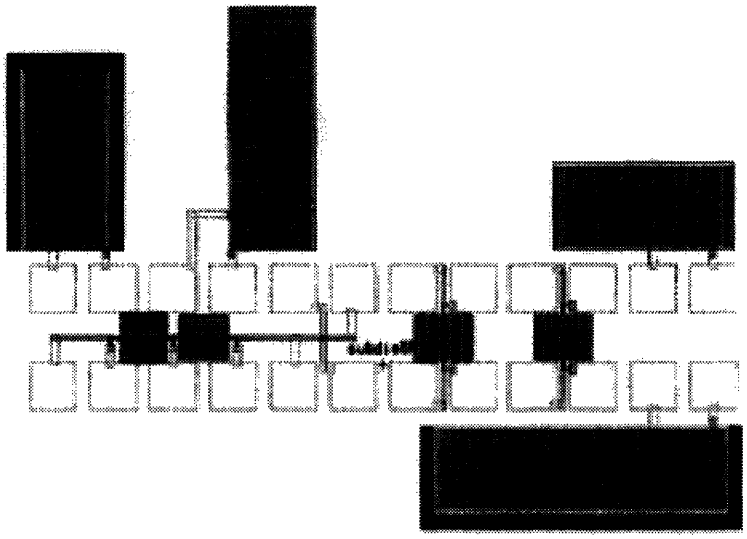

Figure A-4: Module 4

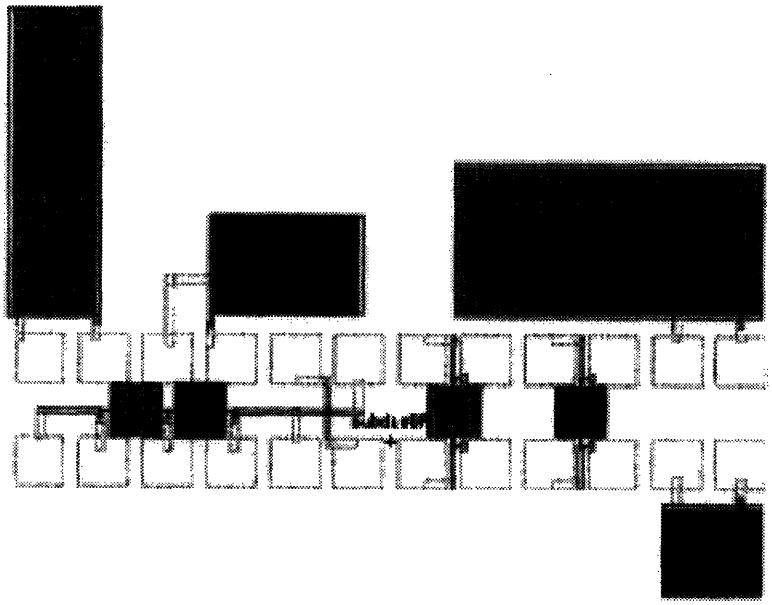

Figure A-5: Module 5

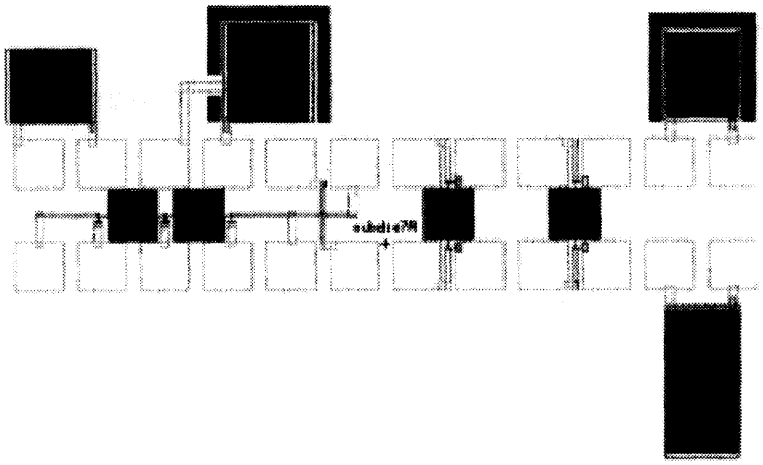

Figure A-6: Module 6

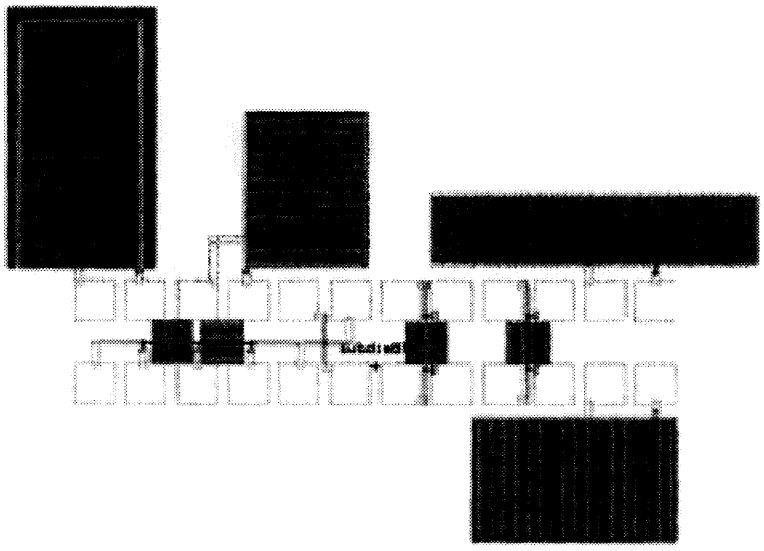

Figure A-7: Module 7

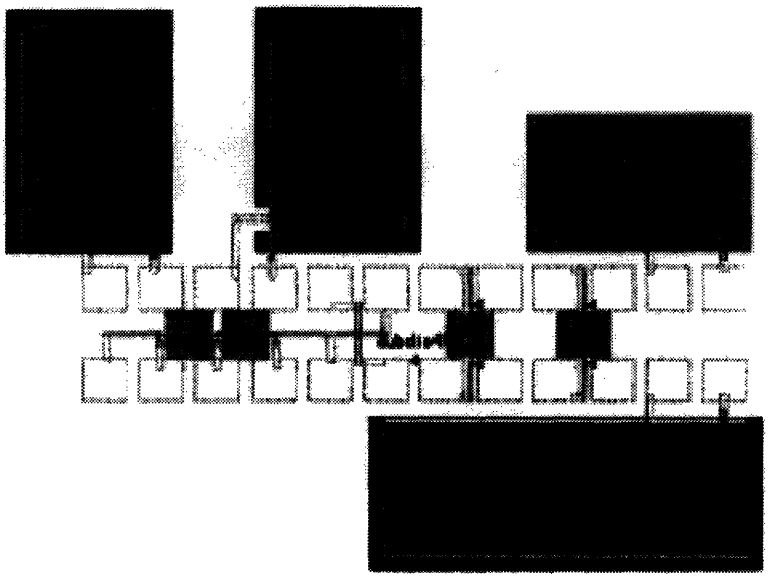

Figure A-8: Module 8

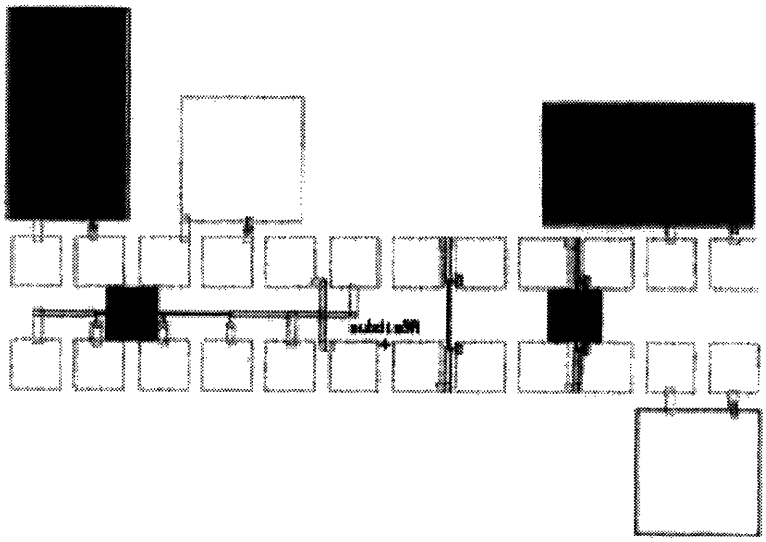

Figure A-9: Module 9

Appendix B

Raphael Input File

```
************* ILD simulation ***********************
****************** Daniel Maung, MIT *********************
* FOX = field oxide thickness
* vFOX = 5ield oxide dielectric constant
* tsub = substrate thickness
* pys0;
* pyth = poly silicon thickness
* m1th = metal 1 thickness
* pywid = poly silicon finger width
* ILD1 = ILD1 thickness                                          10
* vILD1 = ILD1 dielectric constant
* dx = poly CD variation
*
param
FOX  = 0.5e-6;
vFOX = 3.9;
tsub = 1e-6;
pyth = 1e-6;
m1th = 1e-6;
pywd = 5e-6;                                                     20
ILD1 = 0.40e-6;
vILD1 = 4;
pitch = 10e-6;
*
param
pysp = pitch - pywd;
s0 = pysp;
w0 = pywd;
t0 = pyth;
t1 = m1th;                                                       30
h1 = ILD1;
*
param
```

```
Ntrace = 3;
Xtotal = (Ntrace-1)*pitch;
Ytotal = t1 + h1 + t0 + FOX + tsub;
Xmid = 0.5*Xtotal;
cy0= 0.5*t0 + FOX + tsub;
*
* Substrate plane                                                    40
box name = planes; volt = 0.0;
w = Xtotal; h = tsub; cx = Xmid; cy = 0.5*tsub;
*
* Field oxide plane
box name = d_layer0; diel = vFOX;
w = Xtotal; h = FOX; cx = Xmid; cy = tsub + 0.5*FOX;
*
* BPSG plane
box name = d_layer1; diel = vILD1;
w = Xtotal; h = ILD1 + t0; cx = Xmid; cy = tsub + FOX + 0.5*(ILD1+t0);   50
*
* Metal 1 plane
box name = plane1; volt = 1.0;
w = Xtotal; h = t1; cx = Xmid; cy = Ytotal - 0.5*t1;
*
param tcy = cy0;
poly name = trace1; volt = 0.0;
0 , tcy - 0.5*t0;
0 , tcy + 0.5*t0;
0.5*w0 , tcy + 0.5*t0;                                                 60
0.5*w0 , tcy - 0.5*t0;
*
param tcy = cy0;
poly name = trace2; volt = 0.0;
Xmid - 0.5*w0, tcy - 0.5*t0;
Xmid - 0.5*w0, tcy + 0.5*t0;
Xmid + 0.5*w0, tcy + 0.5*t0;
Xmid + 0.5*w0, tcy - 0.5*t0;
*
param tcy = cy0;                                                       70
poly name = trace3; volt = 0.0;
Xtotal - 0.5*w0, tcy - 0.5*t0;
Xtotal - 0.5*w0, tcy + 0.5*t0;
Xtotal, tcy + 0.5*t0;
Xtotal, tcy - 0.5*t0;
*
window diel=1.0;
  x1 = 0.0;
  y1 = 0.0;
  x2 = Xtotal;                                                         80
  y2 = Ytotal;
potential
options set_grid = 6000;
          max_iter = 200;
          iter_tol = 1.0e-3;
          max_regrid = 3;
          regrid_tol = 1e-3;
```

```
unit = 1;
```

Appendix C

Capacitance to Thickness Conversion Program

```
/* Last Modified:  8/24/94  10:00 pm */
/* This program matches capacitance data with ild thickness from simulation */

#include <stdio.h>
#include <malloc.h>
#include <stdlib.h>
#include <float.h>
#include <strings.h>
struct record {
  char wafer[20];                                                    10
  int xc;
  int yc;
  int xw;
  int yw;
  int die, subdie;
  double width;
  double capacitance;
  double thickness;
  struct record *next;
  } ;                                                                20

struct convert {
  double width;
  double thickness;
  double capacitance;
  double capconst;
  double cap;
  double charge;
  struct convert *next;
  } ;                                                                30

struct capbar {
  double thickness;
  double charge;
```

```
  struct capbar *next;
};

/*  FUNCTION PROTOTYPES  */

struct record * read_data1(FILE *, struct record *);                    40
struct convert * read_data2(FILE *, struct convert *);
void assign_ptr(FILE *, struct record *, struct convert *);
void lookup(FILE *, struct record *,struct convert *, struct convert **);
void free_memory1(struct record *);
void free_memory2(struct convert *);
double fab(double);
char infile_struct[3];
double capconst;
int finger_length;
FILE *fin3, *fout3;                                                     50
/*************************************************************************/
main(argc, argv)
int argc;
char *argv[];
{
  FILE *fin1, *fin2,  *fout;
  char infile1[45], in[45], infile2[45], infile3[45], outfile1[45];
  struct record * head1;
  struct convert *head2, **rec_ptr;
  int i=0, length;                                                      60
  int division;
  if (argc != 4)
     {
     printf("Usage:  %s data_file  capsim_file  capbar_file \n", argv[0]);
     exit(1);
     }
  strcpy (infile1, argv[1]);
  strcpy (infile2, argv[2]);
  strcpy (infile3, argv[3]);
  strncpy(infile_struct, argv[1], 3);                                   70
  strcpy(outfile1, infile1);
  strcat(outfile1, ".dat");
  printf("Output file:     ");
  puts(outfile1);

  if((fin1 = fopen(infile1, "r")) == NULL ||
          (fin2 = fopen(infile2, "r")) == NULL ||
       (fin3 = fopen(infile3, "r")) == NULL ||
          (fout = fopen(outfile1, "w")) == NULL )
    {                                                                   80
     printf("W arning:  Cannot open all the files!\n");
     exit(1);
    }
/*****************************************/
  printf("Scanning input files...\n");
  head1 = read_data1(fin1, head1);
  head2 = read_data2(fin2, head2);
  assign_ptr(fout, head1, head2);
```

```
printf("Conversion complete!\n");
free_memory1(head1);                                                    90
free_memory2(head2);
fclose(fin1);
fclose(fin2);
fclose(fout);
}
/*****************************************/
struct record * read_data1(FILE *fin1, struct record * head1)
{
 struct record *index, *next_struct;
 head1 = (struct record *) calloc(1,sizeof(struct record));            100
 if (head1 == NULL)
  { printf("Cannot allocate memory for linked list.\n");
   exit(1);
                    }
 index = head1;
 while(fscanf(fin1, "%E", &index->capacitance) != EOF)
  {     fscanf(fin1, "%E  %s %d %d %d %d %d \n", &index->width,
            &index->wafer, &index->die, &index->subdie,
            &index->xc, &index->yc,&index->xw,&index->yw);
        next_struct = (struct record *) calloc(1, sizeof(struct record));  110
        if (next_struct ==NULL)
          {       printf("Cannot allocate memory for next structure.\n");
                  exit(2);
            }           index->next =next_struct;
        index = index->next;
   }
 index->next = NULL;
 return(head1); }
/*********************************************************************/
struct convert * read_data2 (FILE *fin2, struct convert * head2)        120
{
 struct convert *index,*next_struct;
 struct capbar *head3, *index1, *next_struct1;
 double epsi,thickness;
 int i;
 epsi=8.854E-12;
 head3 = (struct capbar *) calloc (1,sizeof(struct capbar));
 head2 = (struct convert *) calloc(1, sizeof(struct convert));
 /*
  * The following loop reads in the simulated capacitance data        130
  * of the two side bars of the capacitor.
  */
 if (head3 == NULL)
  {
   printf("Cannot allocate memory for linked list.\n");
   exit(1);
  }
 index1 = head3;
 while (fscanf(fin3, "%lf", &index1->thickness) != EOF)
  {                                                                     140
   fscanf(fin3, "%E", &index1->charge);
```

```
    next_struct1 = (struct capbar *) calloc(1, sizeof(struct capbar));
    if (next_struct1 == NULL)
        {       printf("Cannot allocate memory for next structure.\n");
                exit(2);
            }
    index1->next =next_struct1;
    index1 = index1->next;
    }                                                                    150
index1->next = NULL;
index1=head3;

/*  The following loop reads in the simulated capacitance data
 *  of the capacitor
 */

if (head2 == NULL)
  {
    printf("Cannot allocate memory for linked list.\n");                 160
    exit(1);
  }
index = head2;
        if (strcmp(infile_struct, "n1_") == 0)
          {finger_length = 15000; capconst=index1->charge * 178.8 * 2 * 1e-6;}
        if (strcmp(infile_struct,  "n2_") == 0)
          {finger_length = 20000; capconst=index1->charge * 238.8 * 2 * 1e-6;}
        if (strcmp(infile_struct,  "n3_") == 0)
          {finger_length = 22500; capconst=index1->charge * 178.8 * 2 * 1e-6;}
        if (strcmp(infile_struct,  "n4_") == 0)                          170
          {finger_length = 35000; capconst=index1->charge * 238.8 * 2  * 1e-6;}
        if (strcmp(infile_struct,  "n5_") == 0)
          {finger_length = 15000; capconst=index1->charge * 460 * 2 * 1e-6;}
        if (strcmp(infile_struct,  "n6_") == 0)
          {finger_length = 20000; capconst=index1->charge * 615 * 2 * 1e-6;}
        if (strcmp(infile_struct,  "n7_") == 0)
          {finger_length = 22500; capconst=index1->charge * 460 * 2 * 1e-6;}
        if (strcmp(infile_struct,  "n8_") == 0)
          {finger_length = 35000; capconst=index1->charge * 615 * 2 * 1e-6;}
        if (strcmp(infile_struct,  "n9_") == 0)                          180
          {finger_length = 15000; capconst=index1->charge * 178.8 * 2 * 1e-6;}
        if (strcmp(infile_struct,  "n10") == 0)
          {finger_length = 20000; capconst=index1->charge * 618.8 * 2 * 1e-6;}
        if (strcmp(infile_struct,  "n11") == 0)
          {finger_length = 22500; capconst=index1->charge * 463.8 * 2 * 1e-6;}
        if (strcmp(infile_struct,  "n12") == 0)
          {finger_length = 35000; capconst=index1->charge * 618.8 * 2 * 1e-6;}
        if (strcmp(infile_struct,  "n13") == 0)
          {finger_length = 15000; capconst=index1->charge * 745 * 2 * 1e-6;}
        if (strcmp(infile_struct,  "n14") == 0)                          190
          {finger_length = 20000; capconst=index1->charge * 995 * 2 * 1e-6;}
        if (strcmp(infile_struct,  "n15") == 0)
          {finger_length = 22500; capconst=index1->charge * 745 * 2 * 1e-6;}
        if (strcmp(infile_struct,  "n16") == 0)
          {finger_length = 35000; capconst=index1->charge * 995 * 2 * 1e-6;}
        if (strcmp(infile_struct,  "n17") == 0)
```

```
        {finger_length = 23375; capconst=index1->charge * 507 * 2 * 1e-6;
         printf("n17 is found \n");}
      if (strcmp(infile_struct, "n18") == 0)
        {finger_length = 35000; capconst=index1->charge * 238.8 * 2 * 1e-6;}
      if (strcmp(infile_struct, "n19") == 0)
        {finger_length = 22500; capconst=index1->charge * 463.8 * 2 * 1e-6;}
      if (strcmp(infile_struct, "n20") == 0)
        {finger_length = 35000; capconst=index1->charge * 618.8 * 2 * 1e-6;}
      if (strcmp(infile_struct, "n21") == 0)
        {finger_length = 22500; capconst=index1->charge * 178.8 * 2 * 1e-6;}
      if (strcmp(infile_struct, "n22") == 0)
        {finger_length = 35000; capconst=index1->charge * 995 * 2 * 1e-6;}
      if (strcmp(infile_struct, "n23") == 0)
        {finger_length = 22500; capconst=index1->charge * 460 * 2 * 1e-6;}
      if (strcmp(infile_struct, "n24") == 0)
        {finger_length = 35000; capconst=index1->charge * 615 * 2 * 1e-6;}
      if (strcmp(infile_struct, "n25") == 0)
        {finger_length = 22500; capconst=index1->charge * 745 * 2 * 1e-6;}
      if (strcmp(infile_struct, "n26") == 0)
        {finger_length = 20000; capconst=index1->charge * 618.8 * 2 * 1e-6;}
      if (strcmp(infile_struct, "n27") == 0)
        {finger_length = 15000; capconst=index1->charge * 178.8 * 2 * 1e-6;}
      if (strcmp(infile_struct, "n28") == 0)
        {finger_length = 20000; capconst=index1->charge * 238.8 * 2 * 1e-6;}
      if (strcmp(infile_struct, "n29") == 0)
        {finger_length = 15000; capconst=index1->charge * 463.8 * 2 * 1e-6;}
      if (strcmp(infile_struct, "n30") == 0)
        {finger_length = 20000; capconst=index1->charge * 615 * 2 * 1e-6;}
      if (strcmp(infile_struct, "n31") == 0)
        {finger_length = 15000; capconst=index1->charge * 745 * 2 * 1e-6;}
      if (strcmp(infile_struct, "n32") == 0)
        {finger_length = 20000; capconst=index1->charge * 995 * 2 * 1e-6;}
      if (strcmp(infile_struct, "n33") == 0)
        {finger_length = 15000; capconst=index1->charge * 460 * 2 * 1e-6;}

while (fscanf(fin2, "%lf", &index->width) != EOF)
 {
    fscanf(fin2, "%lf   %lf", &index->thickness, &index->charge);

       index1=head3;
       {while ((index1->thickness != index->thickness) &&
                (index1->next != NULL))
        index1=index1->next;}

       index->capacitance = index->charge * 1E-6 * finger_length + capconst;
       index->capconst=capconst;
       index->cap=index->charge*1e-6*finger_length;
       index->width = index->width * 1E6;

    next_struct = (struct convert *) calloc(1, sizeof(struct convert));
    if (next_struct==  NULL)
     {
       printf("Cannot allocate memory for next structure.\n");
       exit(2);
```

```
        }
       index->next = next_struct;
       index = index->next;
     }
  index->next = NULL;
  return(head2);
}
/*********************************************************************/
void assign_ptr(FILE * fout, struct record * head1, struct convert * head2)
{                                                                          260
  struct convert *temp = head2, *index=head2, *index1=head2;
  struct convert **rec_ptr;
  int i=0, j=0, num;
  fout3 =fopen(infile_struct,"w");
  while (index1->next != NULL)
    {
     fprintf(fout3, "%2.2f %f %2.6e %2.6e %2.6e\n", index1->width,
                index1->thickness,  index1->cap, index1->capconst,
                index1->capacitance);
      index1=index1->next; }                                               270
  fclose(fout3);

  while (index != NULL)
    {
      index=index->next;
      i++;
    }
  num = i;
  rec_ptr = (struct convert **) calloc(num, sizeof(struct convert *));
                                                                           280
  while (temp != NULL)
    {
      rec_ptr[j] =temp;
      temp = temp->next;
      j++;
    }
  printf("Start conversion\n");

  lookup(fout, head1, head2, rec_ptr);
}                                                                          290
/*********************************************************************/
void lookup(FILE *fout, struct record * head1, struct convert * head2,
            struct convert ** rec_ptr)
{
  struct record  *temp1=head1;
  struct convert *temp2=head2;
  struct convert *temp3=head2;
  int i,j,m,num,num1,num2,flag;
  double slope,pointac,pointat,pointbc,pointcc,pointct,pointdc,pointdt;
  double pointoc,pointot,pointpc,pointpt,upper_limit,lower_limit;          300
  double pointbt;

 /* temp1 contains the data that needs to be converted */
  while (temp1->next != NULL)
```

```
{
 i=0;
 while(temp1->width >= temp2->width)
      {
       temp2 = temp2->next;
       i++;                                                          310
      }
 upper_limit = rec_ptr[i]->width;
 lower_limit = rec_ptr[i-1]->width;
 num=0;
 while (temp2->width == upper_limit)
      {
       num++;
       temp2=temp2->next;
      }
 while (temp3->width !=  lower_limit)                                320
      {
       temp3=temp3->next;
      }
 num1=0;
 while (temp3->width ==  lower_limit)
      {
       num1++;
       temp3=temp3->next;
      }
 j=i;                                                                330
 while (temp1->capacitance >= rec_ptr[j]->capacitance)
      {
       j++;

      }
 pointac=rec_ptr[j]->capacitance;
 pointat=rec_ptr[j]->thickness;
 pointbc=rec_ptr[j-1]->capacitance;
 pointbt=rec_ptr[j-1]->thickness;
 m=i-num1;                                                           340
      while (temp1->capacitance >= rec_ptr[m]->capacitance)
       {
        m++;

       }
 pointcc=rec_ptr[m]->capacitance;
 pointct=rec_ptr[m]->thickness;
 pointdc=rec_ptr[m-1]->capacitance;
 pointdt=rec_ptr[m-1]->thickness;
 pointoc=((upper_limit-temp1->width)/(upper_limit-lower_limit))*pointcc   350
      + ((temp1->width - lower_limit)/(upper_limit-lower_limit))*pointac;
 pointot=((upper_limit - temp1->width)/(upper_limit-lower_limit))*pointct
      + ((temp1->width - lower_limit)/(upper_limit-lower_limit))*pointat;
 pointpc=((upper_limit-temp1->width)/(upper_limit-lower_limit))*pointdc
      + ((temp1->width - lower_limit)/(upper_limit-lower_limit))*pointbc;
 pointpt=((upper_limit - temp1->width)/(upper_limit-lower_limit))*pointdt
      +((temp1->width - lower_limit)/(upper_limit-lower_limit))*pointbt;
 slope = (pointot-pointpt)/(pointoc-pointpc);
```

```
    temp1->thickness = pointot + slope*(temp1->capacitance-pointoc);

    /* interpolate the thickness corresponding to the given capacitance */

   printf( "%3.3E  %2.3f  %2.3f \n",
            temp1->capacitance,
            temp1->thickness, temp1->width);

    fprintf(fout, "%2.3f %3.3E %2.3f  %s %5d %5d %5d %5d \n",
             temp1->thickness, temp1->capacitance,temp1->width,
             temp1->wafer, temp1->die, temp1->subdie, temp1->xc, temp1->yc,
             temp1->xw,temp1->yw);

    /* output the converted data to the output file */
    temp1 = temp1->next;  /* goto the next capacitance to convert */
    temp2=head2;
    temp3=head2;
   }
}
/*****************************************************************/
void free_memory1(struct record * head1)
{
 if(head1 != NULL)
   {
    free_memory1(head1 -> next);
    free(head1);
    }
 }
void free_memory2(struct convert * head2)
{
 if(head2 != NULL)
   {
    free_memory2(head2 -> next);
    free(head2);
    }
 }
```

Appendix D

HP Data Sorting Program

```
/* last modified: 9/20/94  5 pm */
/* Modified by Daniel Maung */
#include <stdio.h>
#include <stdlib.h>
#include <float.h>
#include <string.h>
#include <ctype.h>

/*  GLOBAL VARIABLES  */
FILE *fin1;                                                              10
FILE *file_name[35];
int  die_x[10] =  {0,0,14580,29160,43740,58320,72900,87480,102060,116640};
int  die_y[10] =  {0,0,14580,29160,43740,58320,72900,87480,102060,116640};

int  subdie_x[53] = {0,-10,2380,4780,-10,2380,4780,-10,2380,4780,
-10,2380,4780,7290,9510,11870,7290,9510,11870,7290,9510,11870,7290
,9510,11870,0,2290,4580,0,2290,4580,0,2290,4580,0,2290,4580,7115,
8975,10835,12075,7115,8820,10680,12075,7115,8820,10525,12075,7115,
8975,10680,12075};
                                                                         20
int  subdie_y[53] = {0,12300,12300,12300,10500,10500,10500,8970,
8970,8970,7280,7280,7280,12270,12270,12270,10630,10630,10630,8880,
8880,8880, 7255,7255,7255, 5240,5240,5240,3440,3440,3440,1780,1780,
1780,0,0,0,4805,4805,4805,4805, 3060,3060,3060,3060, 1315,1315,1315,
-430,-430,-430,-430};

int device_x[9] = {835,60,370,1765,1765,370,525,1145,1455};
int device_y[9] = {185,310,310,310,60,185,185,185,185};

int die_no[9][9] = {{0,0,0,0,0,0,0,0,0}, {0,0,1,2,3,4,5,0,0},            30
{0,6,7,8,9,10,11,12,0}, {13,14,15,16,17,18,19,20,21},
{22,23,24,25,26,27,28,29,30}, {31,32,33,34,35,36,37,38,39},
{0,40,41,42,43,44,45,46,0}, {0,0,47,48,49,50,51,0,0},
{0,0,0,52,53,54,0,0,0}};

char wafer_name[10]="HMJ71-02";
```

```
/*  FUNCTION PROTOTYPES  */
void read_write_data1();
void read_write_data2();                                                    40
void read_write_data3();
void read_write_data4();

/*************************************************************************/

main(argc, argv)
int argc;
char *argv[];
{
   char infile1[20];                                                        50
   char outfile1[10]="n1_m2", outfile2[10]="n2_m2";
   char outfile3[10]="n3_m2", outfile4[10]="n4_m2";
   char outfile5[10]="n5_m2", outfile6[10]="n6_m2";
   char outfile7[10]="n7_m2", outfile8[10]="n8_m2";
   char outfile9[10]="n9_m2", outfile10[10]="n10_m2";
   char outfile11[10]="n11_m2", outfile12[10]="n12_m2";
   char outfile13[10]="n13_m2", outfile14[10]="n14_m2";
   char outfile15[10]="n15_m2", outfile16[10]="n16_m2";
   char outfile17[10]="n17_m2", outfile18[10]="n18_m2";
   char outfile19[10]="n19_m2", outfile20[10]="n20_m2";                     60
   char outfile21[10]="n21_m2", outfile22[10]="n22_m2";
   char outfile23[10]="n23_m2", outfile24[10]="n24_m2";
   char outfile25[10]="n25_m2", outfile26[10]="n26_m2";
   char outfile27[10]="n27_m2", outfile28[10]="n28_m2";
   char outfile29[10]="n29_m2", outfile30[10]="n30_m2";
   char outfile31[10]="n31_m2", outfile32[10]="n32_m2";
   char outfile33[10]="n33_m2", outfile34[10]="n34_m2";

   int count=0, length, i;
                                                                            70
   if (argc != 2)
     {
         printf("Usage:  %s data_file\n", argv[0]);
         exit(1);
     }
   strcpy (infile1, argv[1]);
   if ((fin1 = fopen(infile1, "r")) == NULL ||
         (file_name[1] = fopen(outfile1, "w")) == NULL ||
         (file_name[2] = fopen(outfile2, "w")) == NULL ||
         (file_name[3] = fopen(outfile3, "w")) == NULL ||                   80
         (file_name[4] = fopen(outfile4, "w")) == NULL ||
         (file_name[5] = fopen(outfile5, "w")) == NULL ||
         (file_name[6] = fopen(outfile6, "w")) == NULL ||
         (file_name[7] = fopen(outfile7, "w")) == NULL ||
         (file_name[8] = fopen(outfile8, "w")) == NULL ||
         (file_name[9] = fopen(outfile9, "w")) == NULL ||
         (file_name[10] = fopen(outfile10, "w")) == NULL ||
         (file_name[11] = fopen(outfile11, "w")) == NULL ||
         (file_name[12] = fopen(outfile12, "w")) == NULL ||
         (file_name[13] = fopen(outfile13, "w")) == NULL ||                 90
```

```
        (file_name[14] = fopen(outfile14, "w")) == NULL ||
        (file_name[15] = fopen(outfile15, "w")) == NULL ||
        (file_name[16] = fopen(outfile16, "w")) == NULL ||
        (file_name[17] = fopen(outfile17, "w")) == NULL ||
        (file_name[18] = fopen(outfile18, "w")) == NULL ||
        (file_name[19] = fopen(outfile19, "w")) == NULL ||
        (file_name[20] = fopen(outfile20, "w")) == NULL ||
        (file_name[21] = fopen(outfile21, "w")) == NULL ||
        (file_name[22] = fopen(outfile22, "w")) == NULL ||
        (file_name[23] = fopen(outfile23, "w")) == NULL ||                    100
        (file_name[24] = fopen(outfile24, "w")) == NULL ||
        (file_name[25] = fopen(outfile25, "w")) == NULL ||
        (file_name[26] = fopen(outfile26, "w")) == NULL ||
        (file_name[27] = fopen(outfile27, "w")) == NULL ||
        (file_name[28] = fopen(outfile28, "w")) == NULL ||
        (file_name[29] = fopen(outfile29, "w")) == NULL ||
        (file_name[30] = fopen(outfile30, "w")) == NULL ||
        (file_name[31] = fopen(outfile31, "w")) == NULL ||
        (file_name[32] = fopen(outfile32, "w")) == NULL ||
        (file_name[33] = fopen(outfile33, "w")) == NULL ||                    110
        (file_name[34] = fopen(outfile34, "w")) == NULL
)
   {
    printf("ERROR: Cannot open all the files!\n");
    exit(1);
   }
   /*****************************************/
   printf("Scanning input file...\n");
   read_write_data1();
   fclose(fin1);                                                              120
   fopen(infile1, "r");
   read_write_data2();
   fclose(fin1);
   fopen(infile1, "r");
   read_write_data3();
   fclose(fin1);
   fopen(infile1, "r");
   read_write_data4();

   /* close the input and output files */                                     130
   fclose(fin1);
   for(i = 1; i <= 34; i++)
   fclose(file_name[i]);
   printf("\n\n\n\n");
   printf("************ Conversion Complete!  ************\n");
   printf("\n\n\n\n");
   }

/*************************************************************************/
                                                                              140
void  read_write_data1()
{
 int  i, j, k, no_row=50000, size, cap_x, cap_y, wid_x, wid_y, row,
 col, cap_no;
```

```
  char char_string[15000][20];
  char data[15000];
  int field_c[34] = {14,11,96,99,76,48,184,25,93,51,187,28,85,181,59,
     34,127,113,62,68,17,45,42,31,116,79,8,119,102,178,110,65,82,130};
  int field_w[34] = {22,21,106,107,88,56,192,37,105,57,193,38,91,191,
71,40,139,123,72,74,23,55,54,39,124,89,20,125,108,190,122,73,90,140};

  int subdie_no[34] = {1,1,6,6,5,3,11,2,6,3,11,2,5,11,4,2,8,7,4,4,1,3,
                       3,2,7,5,1,7,6,11,7,4,5,8};
  int device_c[34] = {3,2,2,3,1,3,3,1,1,4,4,2,4,2,1,4,1,
                      2,2,4,4,2,1,3,3,2,1,4,4,1,1,3,3,2};
  int device_w[34] = {7,6,6,7,5,7,7,5,5,8,8,6,8,6,5,8,5,6,6,8,8,6,5,
                      7,7,6,5,8,8,5,5,7,7,6};
  fgets(data, no_row, fin1);
    while (fgets(data, no_row, fin1) != NULL) {
    i=0;
    j=0;

    /* strip newline character */
    size = strlen(data);
    data[size-1] = '\0';

    for (k=0; k< size; k++) {

      if (data[k] == ' ') {
          char_string[i][j] = '\0';
          i++;
          j=0;
      }
      char_string[i][j] = data[k];
      j++;
    }
    char_string[i][j]= '\0';
    row = (int)(char_string[5][1]) - 48;
    col  = (int)(char_string[5][3]) - 48;

    if ( char_string[4][8] == wafer_name[7] ) {
              for (cap_no=0; cap_no <= 33; cap_no++)
          {

          if (isdigit(char_string[field_c[cap_no]][2]) != 0 &&
             isdigit(char_string[field_w[cap_no]][2]) != 0 )
{

          fprintf(file_name[cap_no+1], "%s   ",char_string[field_c[cap_no]]);
          /* print capacitance to output file */

          fprintf(file_name[cap_no+1], "%s   ",char_string[field_w[cap_no]]);
          /* print linewidth to output file */

          fprintf(file_name[cap_no+1], "%s   ",char_string[4]);
          /* print wafer name to output file */

          fprintf(file_name[cap_no+1], "%d   ",die_no[col-1][row-1]);
```

```
        /* print die # to output file */
                                                                        200
        fprintf(file_name[cap_no+1], "%d   ",subdie_no[cap_no]);

        cap_x = die_x[row]+subdie_x[subdie_no[cap_no]]+device_x[device_c[cap_no]];
/* x coordinate of capacitor */

        cap_y = die_y[col]+subdie_y[subdie_no[cap_no]]+device_y[device_c[cap_no]];
/* y coordinate of capacitor */

        fprintf(file_name[cap_no+1], "%d   ", cap_x);
        fprintf(file_name[cap_no+1], "%d   ", cap_y);                   210

        wid_x = die_x[row]+subdie_x[subdie_no[cap_no]]+device_x[device_w[cap_no]];
/* x coordinate of linewidth structure */

        wid_y = die_y[col]+subdie_y[subdie_no[cap_no]]+device_y[device_w[cap_no]];
/* y coordinate of linewidth structure */

        fprintf(file_name[cap_no+1], "%d   ", wid_x);
        fprintf(file_name[cap_no+1], "%d\n", wid_y);
           }                                                            220
        }
   }
   }
  }
/******* second capacitor ************/

void  read_write_data2()
{
  int  i, j, k, no_row=50000, size, cap_x, cap_y, wid_x, wid_y, row, col, cap_no;
  char char_string[15000][20];                                          230
  char data[15000];
  int field_c[34] = {286,283,147,150,348,201,235,263,144,204,238,266,357,
232,399,272,133,215,402,408,289,198,195,269,218,351,280,221,153,229,212,
405,354,136};
  int field_w[34] = {294,293,157,158,360,175,243,275,156,176,244,276,363,
242,411,278,141,225,412,414,295,174,173,277,226,361,292,227,159,241,224,
413,362,142};

  int subdie_no[34] = {17,17,9,9,21,12,14,16,9,12,14,16,21,14,24,16,20,
                       13,24,24,17,12,12,16,13,21,17,13,9,14,13,24,21,8};
                                                                        240
  int device_c[34] = {3,2,2,3,1,3,3,1,1,4,4,2,4,2,1,4,3,2,2,4,4,2,1,3,
                          3,2,1,4,4,1,1,3,3,4};
  int device_w[34] = {7,6,6,7,5,7,7,5,5,8,8,6,8,6,5,8,7,6,6,8,8,6,5,7,
                          7,6,5,8,8,5,5,7,7,6};

  fgets(data, no_row, fin1);
    while (fgets(data, no_row, fin1) != NULL) {
    i=0;
    j=0;
                                                                        250
    /* strip newline character */
    size = strlen(data);
```

```
    data[size-1] = '\0';

    for (k=0; k< size; k++) {

      if (data[k] == ' ') {
           char_string[i][j] = '\0';
           i++;
           j=0;                                                    260
      }
      char_string[i][j] = data[k];
      j++;
    }
    char_string[i][j]= '\0';
    row = (int)(char_string[5][1]) - 48;
    col  = (int)(char_string[5][3]) - 48;

    if ( char_string[4][8] == wafer_name[7] ){
    for (cap_no=0; cap_no <= 33; cap_no++)                         270

           {
           if (isdigit(char_string[field_c[cap_no]][2]) != 0 &&
              isdigit(char_string[field_w[cap_no]][2]) != 0 )
              {
           /* printf("subdie # %d\n", subdie_no[cap_no]); */
           /* printf("%s\t",char_string[field_no[cap_no]]); */
           fprintf(file_name[cap_no+1], "%s   ",char_string[field_c[cap_no]]);
           fprintf(file_name[cap_no+1], "%s   ",char_string[field_w[cap_no]]);
           fprintf(file_name[cap_no+1], "%s   ",char_string[4]);   280
           fprintf(file_name[cap_no+1], "%d   ",die_no[col-1][row-1]);
           /* print die # to output file */

           fprintf(file_name[cap_no+1], "%d   ",subdie_no[cap_no]);

/*         printf("subdie coordinate %d \n", subdie_x[subdie_no[cap_no]]);
           printf(" subdie_no[cap_no] %d \n", subdie_no[cap_no]); */
           cap_x = die_x[row]+subdie_x[subdie_no[cap_no]]+device_x[device_c[cap_no]];

           cap_y = die_y[col]+subdie_y[subdie_no[cap_no]]+device_y[device_c[cap_no]];   290

           fprintf(file_name[cap_no+1], "%d   ", cap_x);
           fprintf(file_name[cap_no+1], "%d   ", cap_y);

           wid_x = die_x[row]+subdie_x[subdie_no[cap_no]]+device_x[device_w[cap_no]];

           wid_y = die_y[col]+subdie_y[subdie_no[cap_no]]+device_y[device_w[cap_no]];

           fprintf(file_name[cap_no+1], "%d   ", wid_x);
           fprintf(file_name[cap_no+1], "%d\n", wid_y);            300
           }
           }
    }
    }
  }
/********** third capacitor  *****************/
```

```
void  read_write_data3()
{
  int  i, j, k, no_row=50000, size, cap_x, cap_y, wid_x, wid_y, row, col, cap_no;
  char char_string[15000][20];
  char data[15000];
  int field_c[34] = {388,385,300,303,365,167,507,416,297,170,510,419,374,504,
450,425,331,317,453,459,391,164,161,422,320,368,382,323,306,501,314,456,371,340};
  int field_w[34] = {396,395,310,311,377,209,515,428,309,210,516,429,380,514,462,
431,343,327,463,465,397,208,207,430,328,378,394,329,312,513,326,464,379,344};
  int subdie_no[34] = {23,23,18,18,22,10,30,25,18,10,30,25,22,30,27,25,20,19,27,
27,23,10,10,25,19,22,23,19,18,30,19,27,22,20};
  int device_c[34] = {3,2,2,3,1,3,3,1,1,4,4,2,4,2,1,4,1,2,2,4,4,2,1,3,3,
                      2,1,4,4,1,1,3,3,2};
  int device_w[34] = {7,6,6,7,5,7,7,5,5,8,8,6,8,6,5,8,5,6,6,8,8,6,5,7,7,
                      6,5,8,8,5,5,7,7,6};

  fgets(data, no_row, fin1);
    while (fgets(data, no_row, fin1) != NULL) {
    i=0;
    j=0;


    /* strip newline character */
    size = strlen(data);
    data[size-1] = '\0';

    for (k=0; k< size; k++) {

      if (data[k] == ' ') {
           char_string[i][j] = '\0';
           i++;
           j=0;
      }
      char_string[i][j] = data[k];
      j++;
    }
    char_string[i][j]= '\0';
    row = (int)(char_string[5][1]) - 48;
    col  = (int)(char_string[5][3]) - 48;

    if ( char_string[4][8] == wafer_name[7]) {
    for (cap_no=0; cap_no <= 33; cap_no++)
           {
           if (isdigit(char_string[field_c[cap_no]][2]) != 0 &&
               isdigit(char_string[field_w[cap_no]][2]) != 0 )
               {
           fprintf(file_name[cap_no+1], "%s   ",char_string[field_c[cap_no]]);
           fprintf(file_name[cap_no+1], "%s   ",char_string[field_w[cap_no]]);
           fprintf(file_name[cap_no+1], "%s   ",char_string[4]);
           fprintf(file_name[cap_no+1], "%d   ",die_no[col-1][row-1]);
           /* print die # to output file */

           fprintf(file_name[cap_no+1], "%d   ",subdie_no[cap_no]);
```

```
        cap_x = die_x[row]+subdie_x[subdie_no[cap_no]]+device_x[device_c[cap_no]];

        cap_y = die_y[col]+subdie_y[subdie_no[cap_no]]+device_y[device_c[cap_no]];

        fprintf(file_name[cap_no+1], "%d   ", cap_x);
        fprintf(file_name[cap_no+1], "%d   ", cap_y);
        wid_x = die_x[row]+subdie_x[subdie_no[cap_no]]+device_x[device_w[cap_no]];

        wid_y = die_y[col]+subdie_y[subdie_no[cap_no]]+device_y[device_w[cap_no]];

        fprintf(file_name[cap_no+1], "%d   ", wid_x);
        fprintf(file_name[cap_no+1], "%d\n", wid_y);
        }
        }
   }

   }
  }

/*********************************************************************/
void  read_write_data4()
{

  int  i, j, k, no_row=50000, size, cap_x, cap_y, wid_x, wid_y, row, col, cap_no;
  char char_string[15000][20];
  char data[15000];
  int field_c[34] = {524,521,470,473,552,252,575,535,467,255,578,538,561,572,586,
544,337,436,589,595,527,249,246,541,439,555,518,442,476,569,433,592,558,334};
  int field_w[34] = {532,531,480,481,564,260,583,547,479,261,584,548,567,582,598,
550,345,446,599,601,533,259,258,549,447,565,530,448,482,581,445,600,566,346};
  int subdie_no[34] = {31,31,28,28,33,15,34,32,28,15,34,32,33,34,35,32,20,26,35,
35,31,15,15,32,26,33,31,26,28,34,26,35,33,20};
  int device_c[34] = {3,2,2,3,1,3,3,1,1,4,4,2,4,2,1,4,3,2,2,4,4,2,1,3,3,2,1,4,
4,1,1,3,3,4};
  int device_w[34] = {7,6,6,7,5,7,7,5,5,8,8,6,8,6,5,8,7,6,6,8,8,6,5,7,7,6,5,8,
8,5,5,7,7,6};

  fgets(data, no_row, fin1);
    while (fgets(data, no_row, fin1) != NULL) {
    i=0;
    j=0;
    /* strip newline character */
    size = strlen(data);
    data[size-1] = '\0';

    for (k=0; k< size; k++) {

      if (data[k] == ' ') {
          char_string[i][j] = '\0';
          i++;
          j=0;
      }
      char_string[i][j] = data[k];
```

```
  j++;
 }
 char_string[i][j]= '\0';
 row = (int)(char_string[5][1]) - 48;
 col  = (int)(char_string[5][3]) - 48;
                                                                          420
 if ( char_string[4][8] == wafer_name[7]) {
 for (cap_no=0; cap_no <= 33; cap_no++)
        {
        if (isdigit(char_string[field_c[cap_no]][2]) != 0 &&
            isdigit(char_string[field_w[cap_no]][2]) != 0 )
            {
        fprintf(file_name[cap_no+1], "%s   ",char_string[field_c[cap_no]]);
        fprintf(file_name[cap_no+1], "%s   ",char_string[field_w[cap_no]]);

        fprintf(file_name[cap_no+1], "%s   ",char_string[4]);                 430

        fprintf(file_name[cap_no+1], "%d   ",die_no[col-1][row-1]);
        /* print die # to output file */

        fprintf(file_name[cap_no+1], "%d   ",subdie_no[cap_no]);

        cap_x = die_x[row]+subdie_x[subdie_no[cap_no]]+device_x[device_c[cap_no]];

        cap_y = die_y[col]+subdie_y[subdie_no[cap_no]]+device_y[device_c[cap_no]];
        fprintf(file_name[cap_no+1], "%d   ", cap_x);                         440
        fprintf(file_name[cap_no+1], "%d   ", cap_y);

        wid_x = die_x[row]+subdie_x[subdie_no[cap_no]]+device_x[device_w[cap_no]];

        wid_y = die_y[col]+subdie_y[subdie_no[cap_no]]+device_y[device_w[cap_no]];
        fprintf(file_name[cap_no+1], "%d   ", wid_x);
        fprintf(file_name[cap_no+1], "%d\n", wid_y);

        }
        }                                                                 450
  }
  }
}
```

460