Copyright © 1991, by the author(s).
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

M91/55
55 Pages

MANUFACTURING-BASED IC PROCESS AND DEVICE SIMULATION

by

Tom L. Luan

Memorandum No. UCB/ERL M91/55

23 May 1991

MANUFACTURING-BASED IC PROCESS AND DEVICE SIMULATION

by

Tom L. Luan

Memorandum No. UCB/ERL M91/55

23 May 1991

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

MANUFACTURING-BASED IC PROCESS AND DEVICE SIMULATION

by

Tom L. Luan

Memorandum No. UCB/ERL M91/55

23 May 1991

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

Manufacturing-based IC Process and Device Simulation

Tom L. Luan
Electronics Research Laboratory
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley, CA 94720

May 23, 1991

ABSTRACT

Traditional Technology Computer Aided Design (TCAD) frameworks lack the ability to precisely describe the idiosyncrasies of a specific fabrication line. This is because the various fabrications steps are simulated with the help of generic models, without the benefit of distinguishing between individual fabrication equipment. In this report, a manufacturing-based TCAD simulation framework is presented. More specifically, the SIMPL-IPX TCAD simulation system has been integrated with the Berkeley Computer-Aided Manufacturing (BCAM) system which supports a public, object-oriented library of statistically based equipment models. In this integrated modular framework, a designer can use TCAD tools, such as the process and device simulators which are available in the SIMPL-IPX system, as well as CAM utilities, such as a recipe editor, a recipe preview and optimization function, a statistical worst case analysis algorithm, and equipment models that are always updated to reflect the aging effects of manufacturing equipment.

Table of Contents

CHAPTER 1 Introduction 1

1.1 History and Overview 1

1.2 Report Organization 3

CHAPTER 2 The BCAM/SIMPL-IPX Modular Framework 5

2.1 The BCAM Equipment Model Library 5

2.1.1 Overview 5
2.1.2 Currently available BCAM Models 7

2.2 The SIMPL-IPX TCAD Simulator 9

2.2.1 Overview 9
2.2.2 Applications and Limitations 10

2.3 Modular Connection between BCAM and SIMPL-IPX 11

2.3.1 BCAM/SIMPL Architecture 12
2.3.2 Implementation Example 12

CHAPTER 3 The BCAM Simulation Utilities 15

3.1 Introduction 15

3.2 The Recipe Editor 15

3.2.1 Editing, Storage and Retrieval 15
3.2.2 X window Forms 16

3.3 Recipe Preview and Optimization 18

3.3.1 The Recipe Previewer 18
3.3.2 Recipe Optimization 18

3.4 The Worst Case Analysis Algorithm 20

3.4.1 Motivation 20
3.4.2 Algorithm Design 21
3.4.3 Implementation 22
3.4.4 X Window Interface 23

CHAPTER 4 Application Examples 25

4.1 Introduction 25

4.2 Simulation Example of a Simple Process Flow 25

4.2.1 Using the LPCVD Equipment Model 25
4.2.2 Using the Plasma Etch Model 27

4.2.3 Example of Worst Case Analysis... 28

CHAPTER 5 Conclusion 30

5.1 Summary of the Report.. 30

5.2 Remaining Problems and Future Works... 30

References.. 31

Appendix A: A List of source code modifications in the SIMPL-IPX system.............................. 33

Appendix B: Worst Case Analysis Interface for LPCVD.. 34

Acknowledgments

I thank my research advisor Professor C. J. Spanos for his valued guidance and support during the entire development of this work. I would also like to thank Professor A. R. Neureuther for reading this thesis and for the helpful discussions.

I would like to express my special appreciations to Haifang Guo, Hao-Cheng Liu, and Sherry Lee for their friendship, encouragement, and sharing of technical expertise. I thank the remaining BCAM members: S. Bana, B. Bombay, E. Boskin, R. Chen, C. Hegarty, S. Leang, G. May, J. Thompson, C. Wang, and previous BCAM members: N. Chang, K. Lin, Z. Ling, and C. Williams. Their contributions truly make BCAM a working group.

During the early stage of this work, E. Scheckler, A. Wong, and A. Lai provided valuable help on understanding and using the SIMPL-IPX simulator. I wish to thank them here. I also thank B. Bombay and S. Lee for proofreading this thesis and the paper resulted from this research project.

I would like to thank all my friends in the EECS department, including Weijie Yun, Yong Liu, Jianhui Huang, Xin Wu, Greg Uehara, Charmine Tung, Yan Zhao, Jian Chen, and William Lam. I have truly enjoyed their friendship and view of life in general. The special friendship from Rick Ye, Boqi Wang, Andrew Block, Ellen Peng, Jianhua Chen, Weixing Wang, and Jon Marr has been very important to me during the last few years. I am also indebted to Kara Yeung for her caring, support and continuous friendship.

The financial support from the SRC (No. 85-04-058) is gratefully acknowledged.

Finally, I would like to thank my family for their love, encouragement, and support.

I dedicate this report to my mother and father.

Chapter 1

Introduction

1.1 History and Overview

Recent efforts have been focusing on creating integrated frameworks which allow a process designer to access a number of computer-aided tools through one unified environment [1] [2]. Technology CAD (TCAD) process and device simulators, such as SIMPL [3], SAMPLE [4], SUPREM [5], and FABRICS [6], can greatly facilitate the development of new technologies. TCAD frameworks and their simulators, however, cannot describe specific fabrication lines, mainly because they lack the ability to describe the idiosyncratic behavior of individual equipment. One way to remedy this shortcoming is to equip a TCAD framework with links that allow direct access to the manufacturing environment. This paper presents a simulation system that incorporates a TCAD framework together with a Computer-Aided Manufacturing system (CAM). More specifically, the workstation based SIMPL-IPX framework [2], which has internal simulation programs and a modular structure to call external simulators, has been linked to the Berkeley CAM system, which offers utilities such as a recipe management system, a recipe preview and optimization function, a worst case analysis statistical algorithm, and several up to date, statistically-based equipment models. Hence, the process developer can switch freely between TCAD and CAM operations in order to perform process and device simulation as well as to evaluate the manufacturability of the proposed structures.

The idea of incorporating manufacturing equipment models in TCAD simulation was introduced as early as 1986 by Hughes and Shott [7]. That vision included equipment models that would be required to supplement the library of process models of existing TCAD systems. Such an architecture, however, is problematic, since it would require a continuous effort to supply and update the equipment models in order to describe today's semiconductor factory. The correct way, in our opinion, is to directly use the equipment models developed and maintained in the manufacturing domain rather than supplement the library of process models of the TCAD systems. This is because the manufacturing equipment models are routinely developed in order to support fabrication applications such as process characterization, equipment maintenance, diagnosis, control, quality and reliability analysis, etc. These models are continuously updated to represent the true status of the equipment. In order to bridge the design and the manufacturing domains, application specific interfaces need to be developed. In 1990, MacDonald *et al.* integrated a process simulator into a commercial CAM system [8]. In that implementation, the simulator was used by the manufacturing engineers for on-line process control. Although this system provides physical insight to manufacturing engineers, it does not employ equipment models and therefore it cannot offer information about manufacturing equipment to process designers.

As mentioned above, the equipment models are developed for manufacturing purposes. They are usually empirical in nature and are derived through extensive, statistically designed experiments. In some implementations, these models are adaptive, i.e. they can easily be modified to reflect the aging effects in equipment. Furthermore, these manufacturing-based equip-

ment models have been created to relate controllable parameters, such as equipment settings, to performance measures which are recorded during production. Typically, the models consist of simple sets of regression equations and include statistical information such as the run-to-run process fluctuations, the across wafer and across lot uniformities, etc. Such equipment models are usually supported by Computer-Integrated Manufacturing (CIM) databases, recipe editors and automated data acquisition systems. Therefore, the integration of equipment models into a TCAD framework is better undertaken not by merely adding some models to the model library of a process simulator, but rather by making the manufacturing-based models, along with their utilities, available to the process designer. In this way, the significant information gap that exists between development and production will be bridged.

This work focuses on making manufacturing-based equipment models available to the TCAD simulation environment. This is accomplished by linking the Berkeley CAM system (BCAM) to the SIMPLE-IPX TCAD framework. Figure 1.1 depicts this connection schematically.

1.2 Report Organization

Chapter 2 briefly reviews the development of the BCAM equipment model library and describes in detail the design and implementation of the modular connection between SIMPL-IPX and the BACM equipment models. Chapter 3 presents the BCAM simulation interface and describes its available utilities. Application examples of a simple process flow simulated using the SIMPL-IPX /BCAM system are given in Chapter 4. Finally, Chapter 5 summarizes the results and discusses remaining problems and future plans.

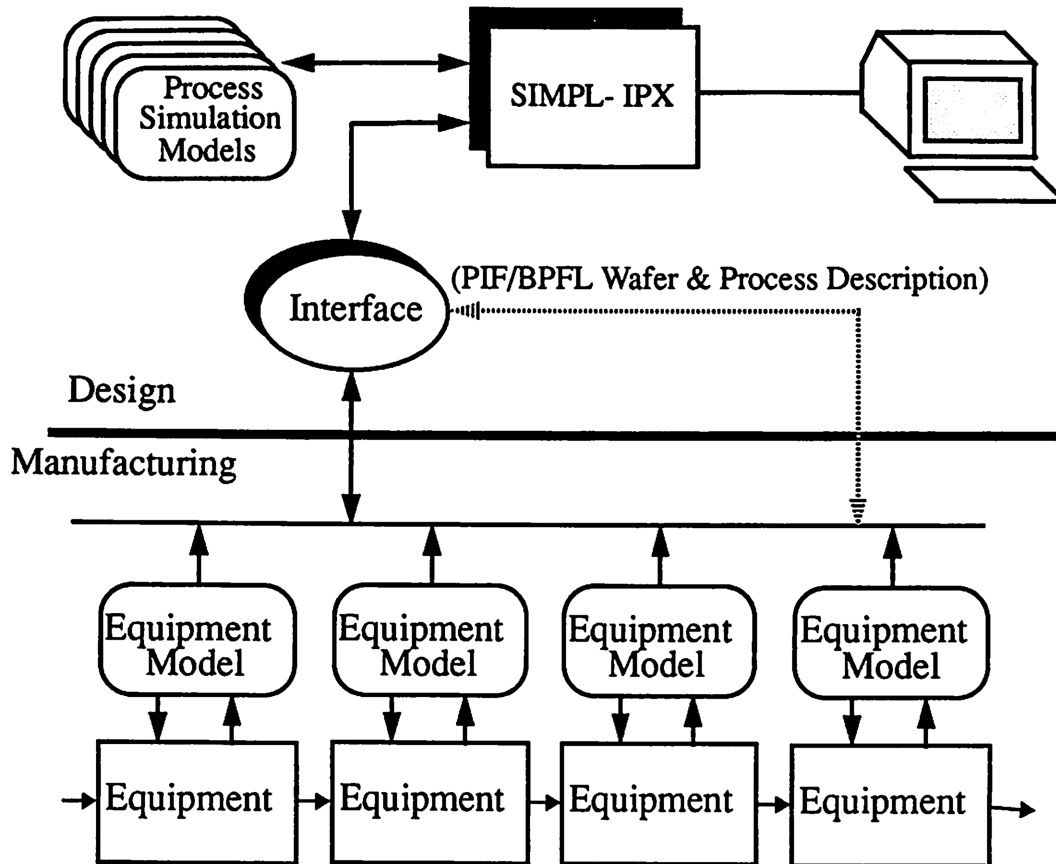

Figure 1.1 The BCAM-SIMPL link

Chapter 2

The SIMPL-IPX/BCAM Framework

2.1 The BCAM Equipment Model Library

2.1.1 Overview

The BCAM system is a workstation-based CAM system built to streamline the operation of semiconductor manufacturing equipment. The BCAM system takes advantage of the object-oriented features of the CLOS and C++ programming languages. It also uses the X-window system and interfaces with a relational database and several commercial statistical packages. It communicates directly with semiconductor equipment through the SECSII protocol. The system is designed to support inter-equipment control in workcell configurations and is a subset of the greater Berkeley Computer-Integrated Manufacturing (BCIM) system. Currently, the BCAM system supports real-time monitoring, diagnosis, recipe generation and management, and modeling. Several process steps are represented, including low-pressure chemical vapor deposition (LPCVD) and plasma etching.

BCAM supports a public equipment model library. These models include LPCVD and plasma etching of polysilicon, as well as spin-coat and bake, exposure and development of photoresist. These models have been built through extensive, statistically designed experiments and are used for run by run equipment control, model-based diagnosis, recipe generation etc. With minor interface programming effort, the same models can be used in the TCAD environment for process design and simulation. This work presents the incorporation of the

BCAM equipment models and the related BCAM utilities to the SIMPL-IPX TCAD framework.

The BCAM equipment models are organized in an object-oriented library implemented in C++. This implementation takes advantage of the object-oriented properties of hierarchy, inheritance and modularity. It allows the equipment models to reside on a number of compute and file servers, since those models are created, tuned and updated in conjunction with the actual equipment they represent. Further, the equipment models in the library can be used across the network through remote procedure calls. Figure 2.1 shows the structure of the BCAM equipment model library. Figure 2.2 summarizes the pattern transfer aspects that can be simulated using the BCAM equipment models. The statistical aspects of the equipment models are used in the worst case analysis algorithm which is described in Section 3.4. Examples of using these equipment models in the TCAD framework can be seen throughout the following chapters. What follows is a brief overview of the equipment models that are currently available.

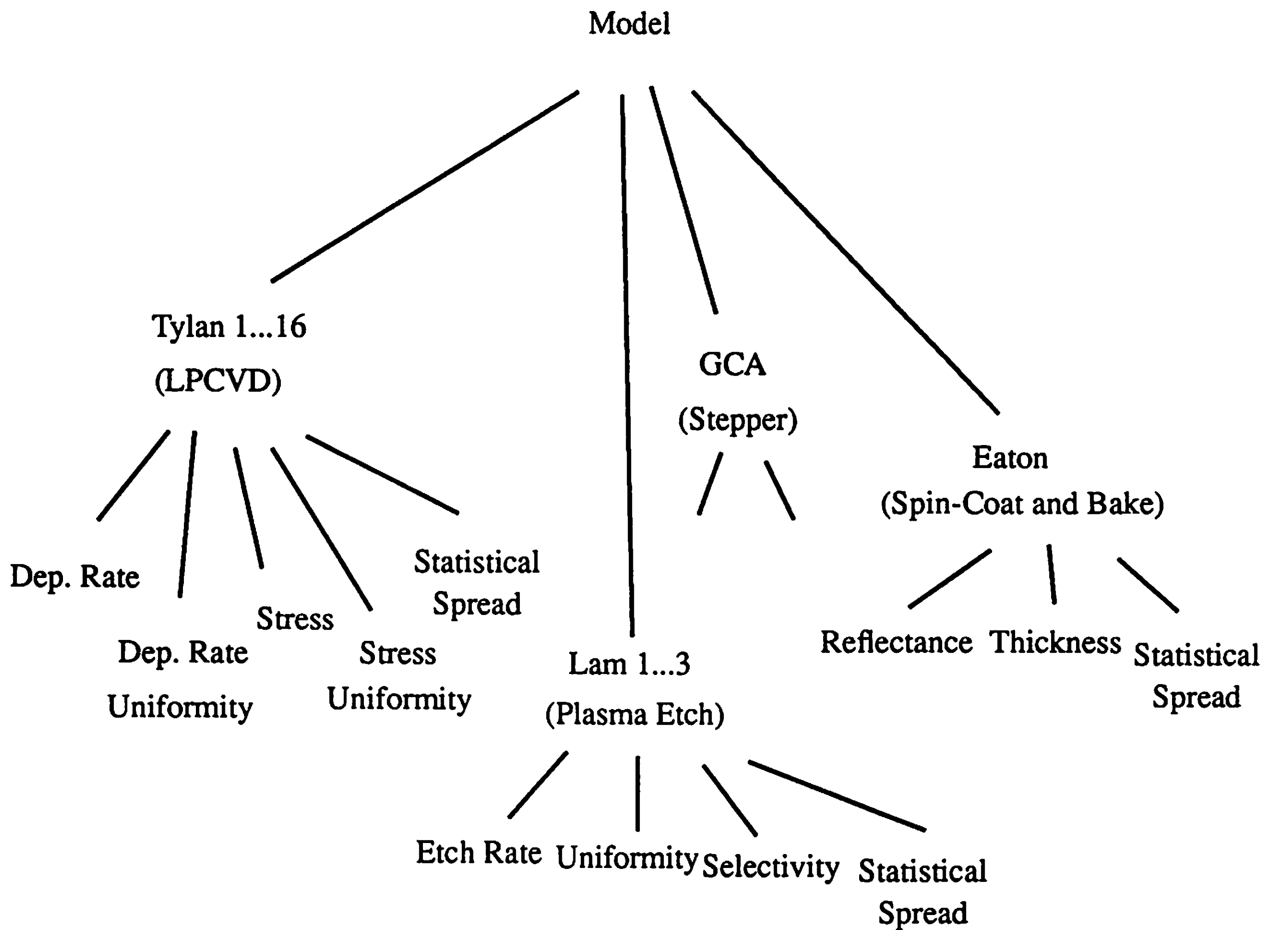

Figure 2.1 The BCAM object-oriented equipment model library

2.1.2 Currently Available BCAM models

LPCVD of Polysilicon: This model represents the operation of a Tylan Low-Pressure Chemical Vapor Deposition Furnace. The model is built using a 2-stage, 24-run D-optimum statistical experiment [9]. The user-defined "Recipe" includes settings for Pressure, Tempera-

ture, Silane flow, and Deposition Time. Given the recipe, the model predicts the Deposition rate (see note [a] in Fig. 2.2), Deposition non-Uniformity, Built-in stress and Built-in Stress non-Uniformity. The model also returns the prediction error of the regression equation and the replication error (run-to-run variability) of the furnace.

Photolithography: A complete photolithographic sequence is modelled using a set of response surfaces through a number of fractional factorial statistical experiments [10]. The models can be used to predict the Thickness (note [b] in Figure 2.2) and the maximum Reflectance of spin-coated photoresist versus the Spin speed, the Spin time, and the Pre-bake temperature for an Eaton wafer track; other models predict the Post-exposure reflectance versus Dose for a GCA stepper; and the Critical dimension ([c] in Figure 2.2) versus the Development time for an MTI developer. These models allow the complete simulation of the photolithographic sequence in the Berkeley Microfabrication Laboratory.

Plasma Etching: This model has been extracted through a 2-stage, 53-run Box-Wilson experiment on a Lam Research Autoetch 490 [11]. The recipe includes the RF power, Pressure, Electrode spacing, as well as the He, CCl_4 and O_2 flows. The model predicts the polysilicon Etch rate, Uniformity, Selectivity to Oxide and to Photoresist, and Etching Anisotropy ([d], [e], [f], and [g] in Figure 2.2, respectively).

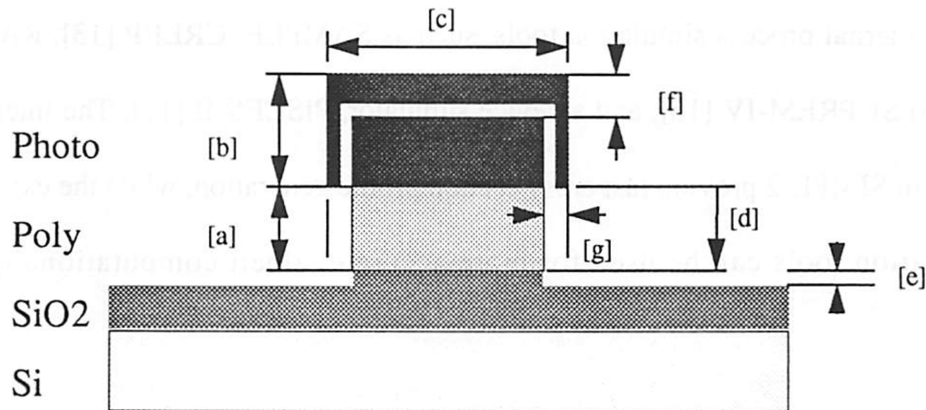

Figure 2.2 Profile modelled with Equipment-specific Expressions

2.2 The SIMPL-IPX TCAD Simulator

2.2.1 Overview

SIMPL-IPX (SIMulation of Profiles from the Layout - Integrated Process simulation with X-windows) is an integrated process simulation environment. SIMPL-IPX consists of two principal components: SIMPL-2 [3] and SIMPL-DIX [12]. SIMPL-2 provides cross-section manipulation and is designed so that when an external simulator is available for a process step, SIMPL-2 can easily be modified to incorporate the external simulator. SIMPL-DIX pro-

vides an X-widow interface and graphic options. Together, SIMPL-IPX allows the user to generate two dimensional cross-sectional profiles from the layout and provides X-window interface for graphic display and command selections, as well as a modular structure to call a variety of external process simulation tools, such as SAMPLE, CREEP [13], RACPLE [14], SPLAT [15] SUPREM-IV [16], and a device simulator, PISCES-II [17]. The internal analytical models in SIMPL-2 provide fast cross-section profile generation, while the external numerical simulation tools can be used for more accurate, albeit computationally intensive predictions.

2.2.2 Applications and limitations

The SIMPL-IPX framework can simulate process steps such as deposition, etch, implantation, oxidation, and re-flow, as well as various lithographic steps such as masked expose and develop. This framework provides good physical insight at the preliminary design phase. It does not allow, however, a detailed manufacturability analysis at the equipment level. One way to remedy this shortcoming is to make equipment models available in the TCAD environment, as discussed in Chapter 1. Below we present the design and implementation of the manufacturing-based SIMPL-IPX/BCAM framework which allows the designer to use equipment models as well as CAM utilities such as a recipe editor, a recipe preview and optimization function, and a worst case analysis statistical algorithm. More details about these CAM utilities are described in Chapter 3.

2.3 Modular Connection between BCAM System and SIMPL-IPX

As mentioned above, the manufacturability of a semiconductor fabrication sequence can be evaluated by using manufacturing equipment models within the TCAD framework. Figure 2.3 illustrates this integrated structure.

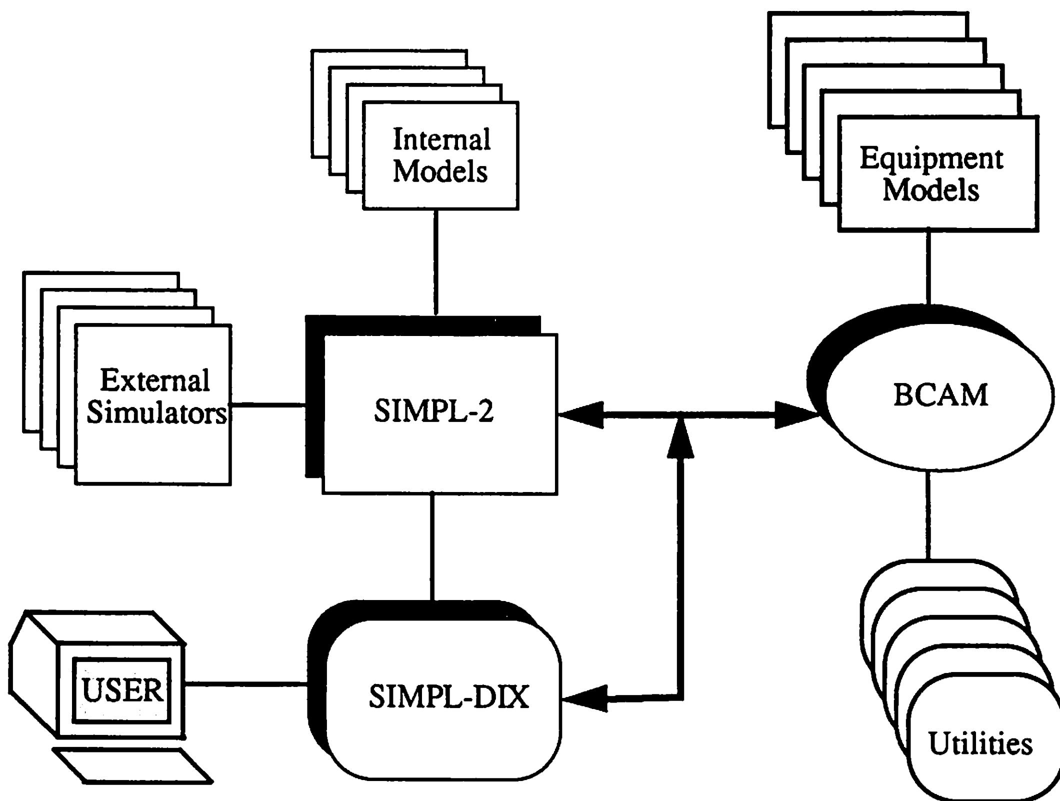

Figure 2.3 The integrated SIMPL/BCAM structure

2.3.1 BCAM/SIMPL Architecture

SIMPL-IPX is chosen as the simulation supervisor due to its modular structure and user-friendly interface, which allows generation of command inputs to external programs with a minimum programming effort (Appendix A lists the SIMPL-IPX source code that has been modified to allow access to the BCAM system). Several modules have been added on the SIMPL-IPX interface to allow the user to call the equipment models by simply clicking the corresponding buttons. Furthermore, some BCAM utilities are coupled into the SIMPL-IPX dialog facilities. An example is given in the next section. The programming efforts needed to add the BCAM equipment models is similar to that of adding additional external simulators to the SIMPL-IPX environment - when there is an available equipment model for a process step, SIMPL-IPX is modified to allow the access to that model. The user is then given a choice to use either a SIMPL internal analytical model, an external simulator, or a manufacturing equipment model. Several interaction examples are given in Chapter 4.

Although this architecture does not benefit from a widely advertized (but not yet implemented) profile interchange standard, it still resulted in a practical prototype of manufacturing-based simulation.

2.3.2 Implementation Example

Figure 2.4 presents an example of the implementation of the BCAM equipment models and utilities into the SIMPL-IPX framework. The "TYLAN16 EQUIP MODEL" addition to the standard SIMPL-IPX menu allows the user to call the respective BCAM model. Alternatively, the user can choose the "SIMPL" button to take the normal SIMPL-IPX path. In the fig-

ure shown in Figure 2.4, the cross-section after growth of oxide simulated by SIMPL models and the deposition of polysilicon using the BCAM LPCVD equipment model. The dialog box shown in the figure gives the user option to invoke the BCAM worst case analysis utility to be discussed in Chapter 3.4.

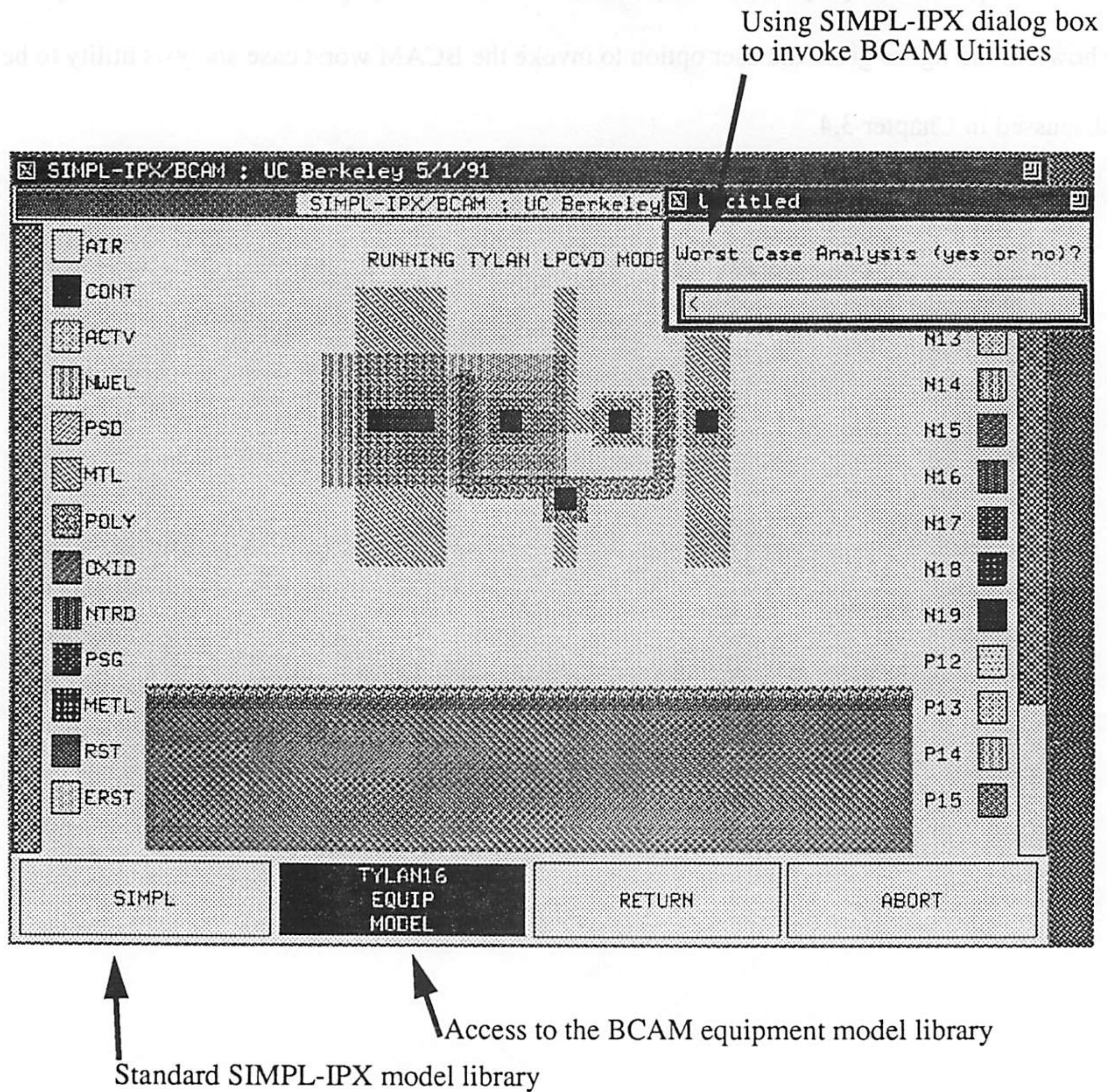

Figure 2.4 Example of incorporating BCAM equipment models and utilities

Chapter 3

The BCAM Simulation Utilities

3.1 Introduction

The BCAM utilities are developed to enhance the utility of the equipment models in the SIMPL-IPX framework. The available BCAM simulation utilities include a recipe editor, a recipe preview and optimization function, as well as a special dialog that allows access to the statistical information contained in the equipment models. Figure 3.1 illustrates the structure of these utilities. Details are given in the following sections.

3.2 The Recipe Editor

3.2.1 Editing, Storage, and Retrieval

The recipe editor allows window-based retrieval, storage and editing of tool settings. The editor acts as a front end to the Ingres recipe database. When the recipe editor is used, a recipe template appears. This template has default settings for the equipment in question. The user can edit the default values or retrieve other recipe versions from the database. The user can also save the recipe for future use. The recipe editor has three dialog buttons. The "Simulate" button sends the recipe and the virtual wafers to the equipment models and the results are returned to the TCAD framework. If the results are satisfactory, the recipe can be stored in the database by the "Save" button and it can later be downloaded directly to the equipment in question. The "Retrieve" button allows the user to examine and use other recipe versions in the database.

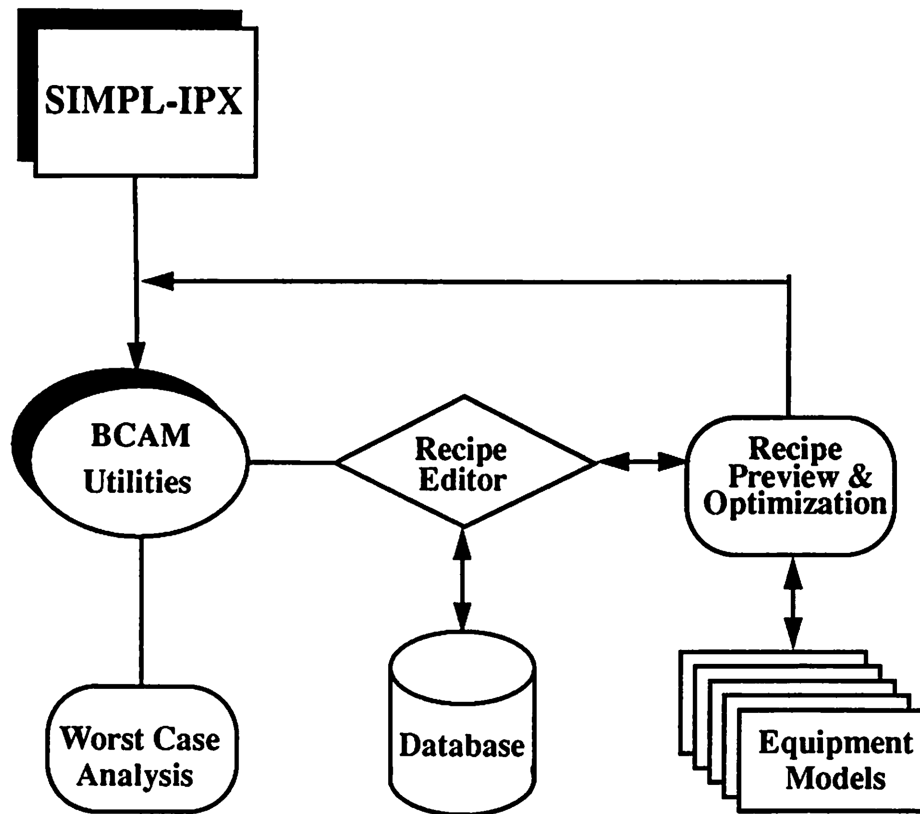

Figure 3.1 The BCAM simulation utilities

3.2.2 X window Forms

The X window system is a hardware independent windowing system adopted by the computer industry [18]. The recipe editor uses the X-window system to display forms that contain the basic recipe template. Figure 3.2 show such X window forms for different equipment models. The boxes that contain numbers are the dialog boxes which allow the user to edit the tool

settings. The three buttons, “Simulate”, “Save”, and “Retrieve”, located on the bottom of the form activate the three callback functions described in the previous section.

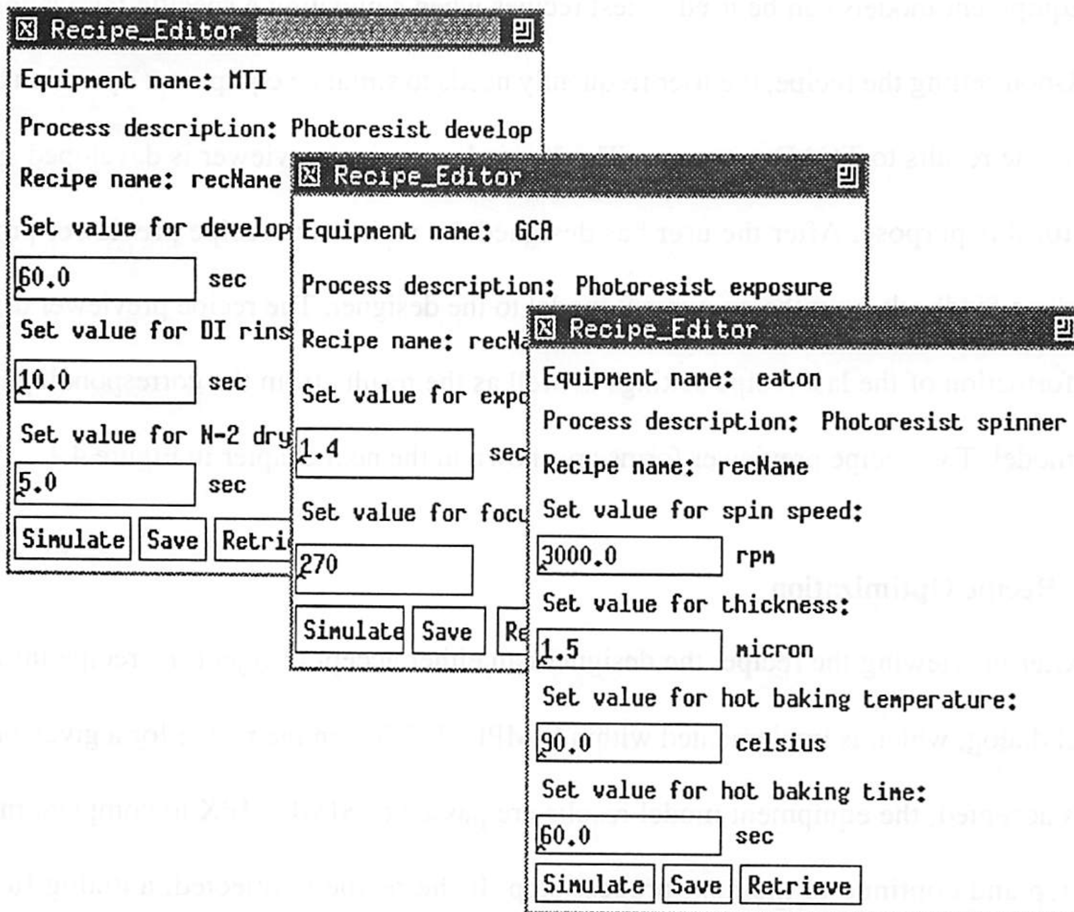

Figure 3.2 X window form recipe editors

3.3 Recipe Preview and Optimization

The recipe preview and optimization function consists of a recipe previewer and a special dialog function that allows the designer to optimize manufacturing recipes within the TCAD simulation environment.

3.3.3 The Recipe Previewer

Equipment models can be used to test recipes when simulating a specific fabrication process. Upon setting the recipe, the user frequently needs to simulate equipment operation before passing the results to TCAD programs. The X-window recipe previewer is developed specifically for this purpose. After the user has designed the recipe, the recipe previewer provides immediate feedbcak from the equipment model to the designer. The recipe previewer displays the information of the last recipe settings as well as the results from the corresponding equipment model. Two recipe previewer forms are shown in the next chapter in Figure 4.1.

3.2.4 Recipe Optimization

After previewing the recipe, the designer can either accept or reject the recipe through a special dialog, which is implemented within SIMPL-IPX. When the recipe for a given process step is accepted, the equipment model results are passed to SIMPL-IPX to complete the current step and continue to the next process step. If the recipe is rejected, a dialog function guides the user back to the recipe design stage. Figure 3.3 illustrates the recipe optimization procedure within the SIMPL-IPX framework.

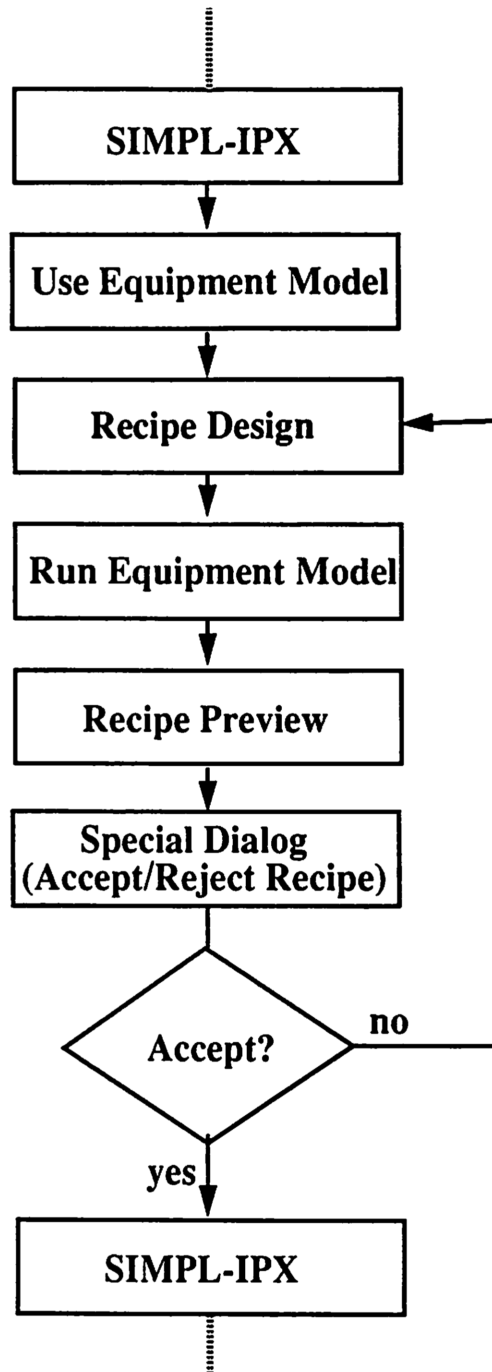

Figure 3.3 The recipe Preview and Optimization sequence

3.4 The Worst Case Analysis Algorithm

3.4.1 Motivation

Equipment models are developed in a manufacturing environment, typically after the completion of a number of experimental runs. Many of these models contain estimates of the run-by-run replication errors of the equipment. Such data may be used to analyze the process variation and to test the manufacturability of a proposed manufacturing sequence. This is accomplished by conducting a worst case analysis of the newly designed process.

Because of the numerous performance aspects of the manufacturing tools, a complete worst case analysis of the simulated process flow would be tedious and complicated. For example, the empirical LPCVD model returns the following parameters:

- Polysilicon Thickness Mean (μ_t) and run-to-run standard deviation (σ_t)
- Thickness Uniformity Mean (μ_{tu}) and run-to-run standard deviation (σ_{tu})
- Built-in Stress Mean (μ_s) and run-to-run standard deviation (σ_s)
- Stress Uniformity Mean (μ_{su}) and run-to-run standard deviation (σ_{su})

Each of these four parameters has an effective statistical range. Since the run-to-run variation of these parameters are independent, each has a 95% confidence interval defined between μ-1.96σ and μ+1.96σ. There are $2^4 = 16$ possible extreme value combinations. Not all of the combinations lead to a worst case. For example, during the simulation of the LPCVD process, the key physical parameter is the polysilicon thickness which is inherited by the next process step. Since we are only interested in the extreme values of thickness, it is not necessary to pass all these 16 combinations to the next step. In that particular case, choosing the extreme values

of the thickness, combined with the highest value of thickness non-uniformity will capture the extreme behavior of the LPCVD process step.

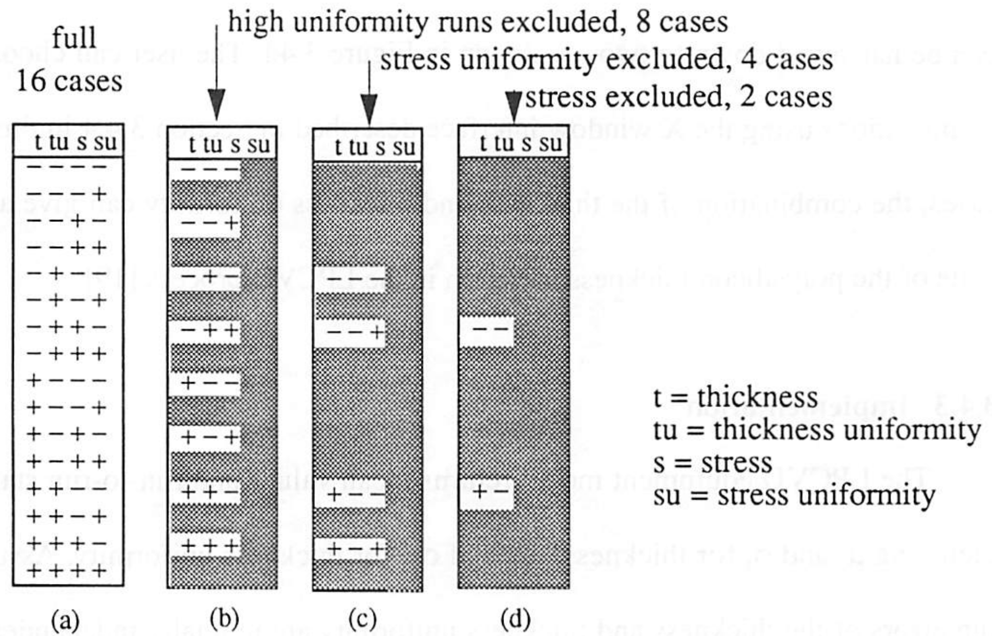

Figure 3.4 Example of LPCVD worst case analysis

3.4.2 Algorithm Design

An interactive interface has been developed to allow the user to choose the relevant subset of the meaningful worst case combinations. Figure 3.4 illustrates this concept. Figure 3.4a shows the total of 16 possible combinations. If the stress uniformity is not important, eliminat-

ing it leaves eight combinations as shown in Figure 3.4b. Examining the physical meaning of thickness uniformity, we find that a high value (highly uniform) does not lead to extreme thickness values. This reduces the number of worst case combinations to four as shown in Figure 3.4c. Furthermore, if the impact of stress is negligible in our design, then the worst cases can be narrowed down to two, as shown in Figure 3.4d. The user can choose these different combinations using the X window interface described in Section 3.4.4 in this chapter. In most cases, the combination of the thickness and thickness uniformity can give a satisfactory estimate of the polysilicon thickness variation in the LPCVD process [19].

3.4.3 Implementation

The LPCVD equipment model returns mean values and run-to-run standard deviations, including μ_t and σ_t for thickness, μ_{tu} and σ_{tu} for thickness uniformity. Assuming the run-to-run errors of the thickness and thickness uniformity are mutually independent, the worst case combinations of the thickness and thickness uniformity for the LPCVD process lead to the following low, nominal, and high thickness values:

$$t_{low} = \mu_t - \kappa_t \cdot \sigma_t - \mu_t \cdot \xi_{max}$$

$$t_{nominal} = \mu_t$$

$$t_{high} = \mu_t + \kappa_t \cdot \sigma_t + \mu_t \cdot \xi_{max}$$

where ξ_{max} is the maximum thickness non-uniformity (in%) given as:

$$\xi_{max} = 1 - (\mu_{tu} - \kappa_{tu} \cdot \sigma_{tu})$$

κ_t and κ_{tu} are constants that are set to = 1.96, in order to obtain a 95% confidence interval.

The relations of the stress and stress uniformity with the thickness are not as straightforward as that of thickness and thickness uniformity. The models of these relations are most likely to be empirically developed through extensive experiments. At the present, these relations are not available in the LPCVD equipment model. However, their callback function paths have been set up on the worst case analysis interface (see Figure 3.5). When these models are available, they can be directly linked to the interface.

3.4.4 X Window Interface

Figure 3.5 shows the worst case analysis interface for the LPCVD equipment model. The three dialog buttons in the middle allow the user to choose the desired worst case components. Only the combination of thickness and thickness uniformity is available now as mentioned above. The "Help" Button on the bottom displays background instructions and information about the interface.

When the "T & TU" button is selected, the callback function calculates the nominal, low and high values of the polysilicon thickness according to the equations listed above. The user can take the nominal value to propagate the process simulation step while saving the low and high values and their associated profiles. At the end of the simulation flow, the user can return

to this step and repeat the remaining steps with either low or high value of polysilicon thickness. This subject is discussed further in Chapter 4.

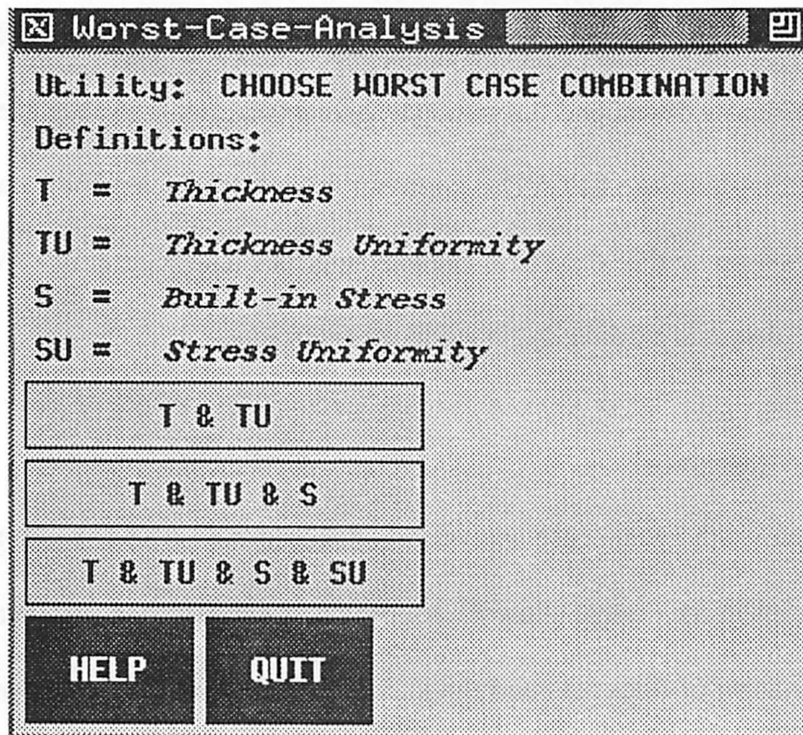

Figure 3.5 Worst case analysis interface for LPCVD

Chapter 4

Application Examples

4.1 Introduction

The purpose of the following examples is to illustrate the operation of a unified framework that brings together a TCAD framework and manufacturing tool models. The user can design, simulate, and define the proposed sequence and resulting structures in terms of equipment performance and their limitations. SIMPL-IPX allows the user to display the cross sectional view of a device along an arbitrary cut-line on the layout of the device. The layout of a CMOS inverter is shown in a SIMPL-IPX window in Figures 4.1 and 4.2. To demonstrate how equipment models are used in simulation, a simulation sequence that defines the gate of the n-channel transistor is presented. First, the user chooses a cut-line across the n-channel gate. Then, the user can interactively simulate deposition, photolithography, etching and other process steps. The cross section profiles of each step are displayed under the layout.

4.2 Simulation Example of a Simple Process Flow

The process flow simulated below is for a n-channel gate. The process flow starts with the growth of oxide on a p-doped silicon wafer. BCAM equipment models are invoked during the simulation of this sequence.

4.2.1 Using the LPCVD Equipment Model

Figure 4.1 shows an example of calling the LPCVD equipment model to simulate polysilicon deposition. When the user clicks the "TYLAN16 EQUIP MODEL" button, a recipe

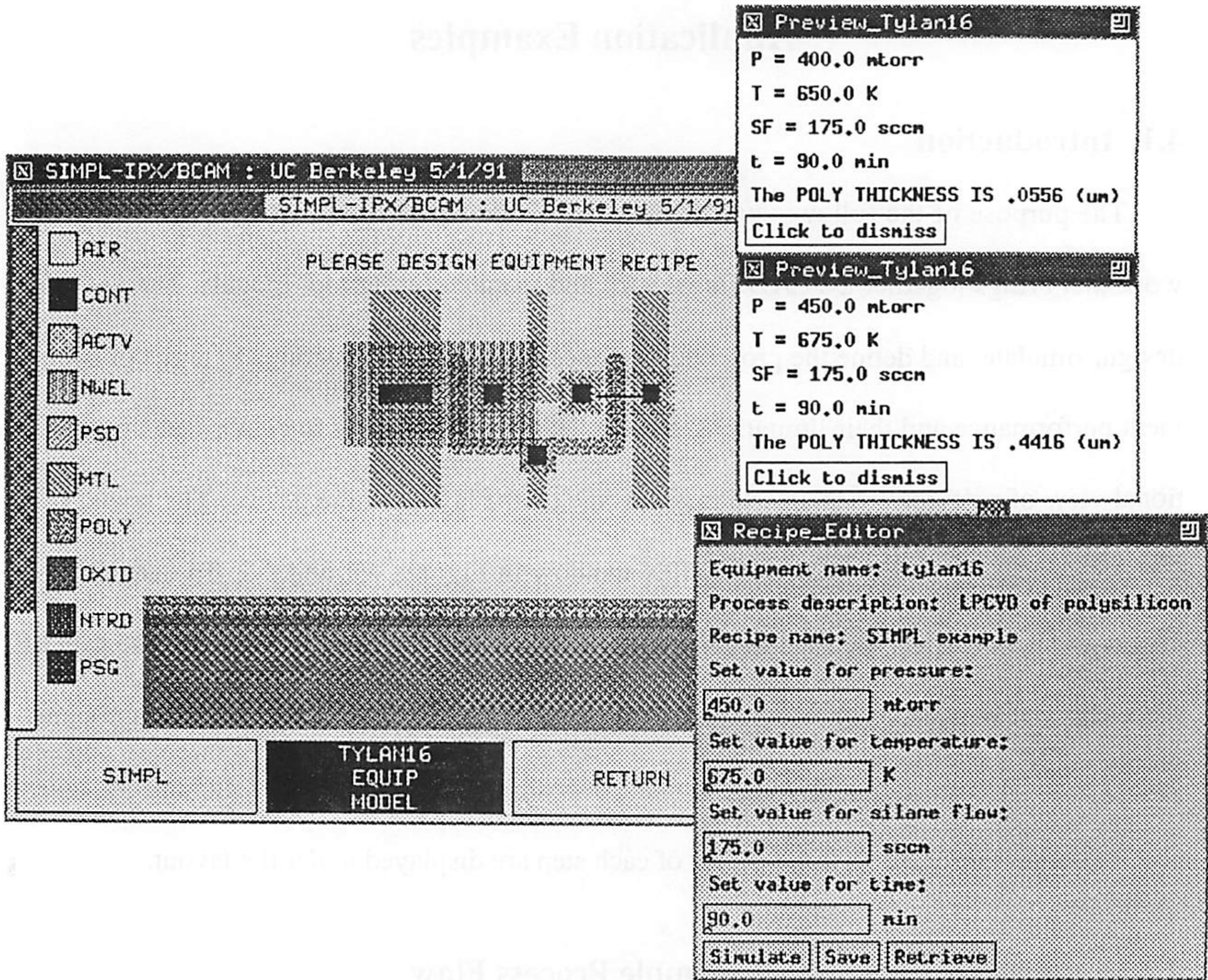

Figure 4.1 Recipe Editor and Previewer in LPCVD

template appears. After designing the recipe, the user clicks the “Simulate” button to invoke the equipment model. The recipe previewer provides a summary of the last recipe as well as the predictions of the equipment model. The special dialog function (not shown) allows the

user to accept or reject the recipe before passing the result, in this case the polysilicon thickness, to the TCAD simulator. In the example shown in Figure 4.1, the final recipe resulted in a layer of polysilicon with a thickness of 0.44μm. This recipe is then stored in the CIM data base and can later be used by the process engineers to implement the simulated process flow.

4.2.2 Using the Plasma Etch Equipment Model

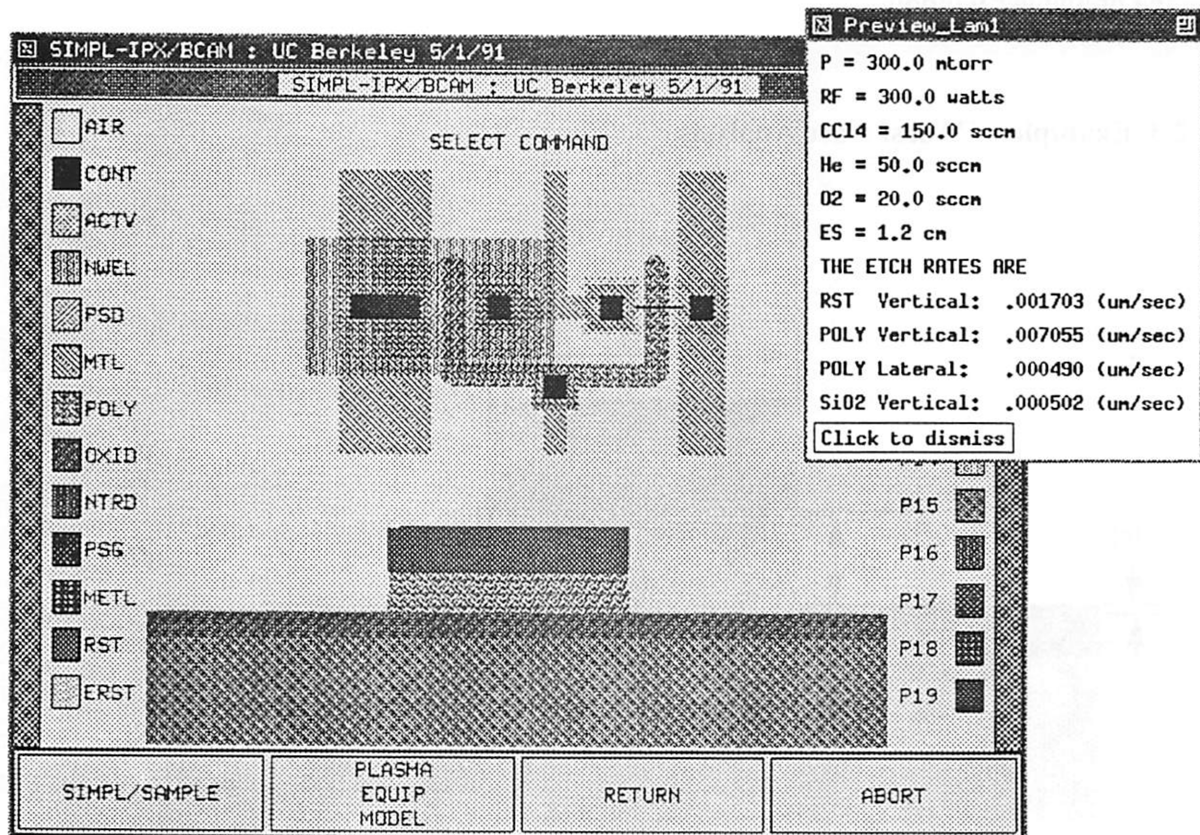

Figure 4.2 Profile of Invoking Equipment Models after Plasma Etch

The profile of the n-channel gate after the completion of the plasma etching step is shown in Figure 4.2. This profile has been obtained by invoking the BCAM equipment models during the simulated flow. This profile has the same layers as the one in Figure 2.2. The amount of the oxide etched in Figure 4.2 is negligible, due to the excellent selectivity of the Lam Research Autoetch 490. Lateral etch rates are also very low, as expected in a commercial dry etch application. A summary of the recipe and the resulting equipment performance can be seen in the recipe previewer window.

4.2.3 Example of Worst Case Analysis

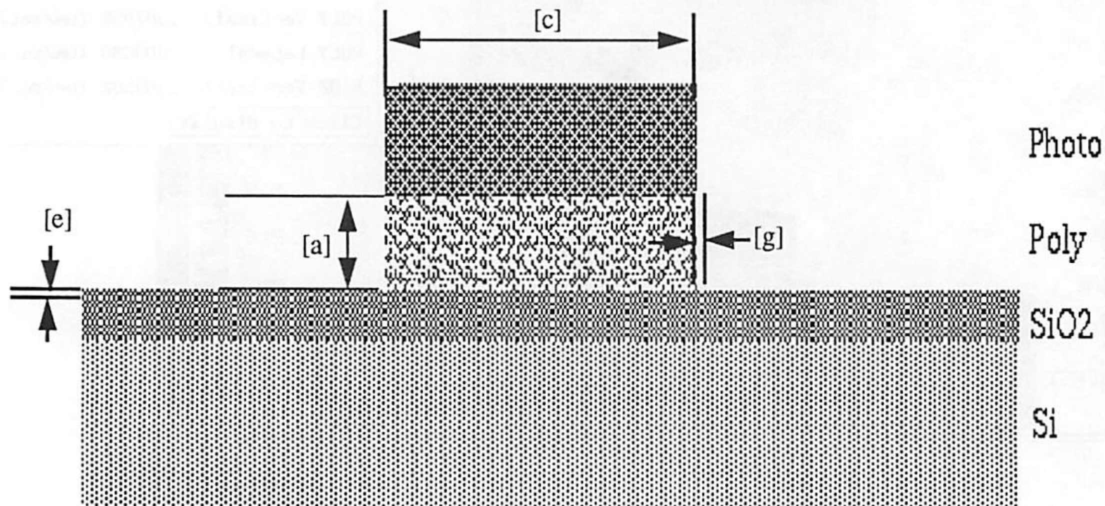

Figure 4.3 Profile to illustrate the worst case analysis

Figure 4.3 shows the actual profile extracted from the cross-sectional view shown in the lower part in Figure 4.2. Assuming the polysilicon is completely etched away outside the photoresist mask area, the n-channel gate polysilicon thickness (dimension [a] in Figure 4.3) variations can be characterized using the worst case analysis algorithm for the LPCVD equipment model described in Chapter 3.4. For the equipment recipe which results in a 0.442μm nominal thickness (see the second recipe previewer in Figure 4.1), the worst case combination of errors from thickness and thickness uniformity returns 0.335μm and 0.548μm as the low and high extremes. Very little lateral poly etch occurred (dimension [g] in Figure 4.3), and its variation is directly determined by the standard deviation of the lateral polysilicon etch rate. The thickness of the removed oxide during the etching (dimension [e] is Figure 4.3) is very small and its variations are negligible.

Chapter 5

Conclusions

5.1 Summary of the Report

This work focuses on the integration of a TCAD framework with manufacturing-based equipment models. Along with the integration, some CAM utilities have been developed to help the process designer to manipulate tool recipes. These utilities include a recipe editor, a recipe preview and optimization function, and a worst case analysis algorithm that takes advantage of the statistical content of the manufacturing-based equipment models. Within this integrated framework, the user can design a process at the equipment recipe level. Therefore the designer takes into consideration the manufacturability of the proposed process and device structures. Since the final outcome of such a design session is a set of tool recipes that can be directly downloaded to the equipment in question. The gap between design and manufacturing is significantly narrowed.

5.2 Remaining Problems and Future Works

A key issue that must be resolved is the efficient representation of wafer and process status. Pioneering work in this field includes the development of the Profile Interchange Format (PIF) [20] [21] and the Berkeley Process Flow Language (BPFL) [22]. To date, however, there exists no widely accepted industrial standard. Fortunately such a standard is now actively pursued [23]. Once such a standard emerges, it will greatly facilitated the expansion of the SIMPL-IPX/BCAM framework presented here.

References

[1] P. LLoyd, H.K. Dirks, E.J. Pendergast, and K. Singhal, "Technology CAD for competitive products", IEEE Transaction on Computer-aided Design, vol 9, no 11, pp1209-1216, November 1990.

[2] E. W. Scheckler, A. S. Wong, R. H. Wang, G. Chin, J. R. Camanga, K.K. H. Toh, K. H. Tadros, R. A. Ferguson, A. R. Neureuther, and R. W. Dutton, "A Utility Based Integrated Process Simulation System", Proceedings of the IEEE Symposium on VLSI Technology, Honolulu, Hawaii, Digest of Technical Papers, pp. 97-98, June 1990.

[3] K. Lee, A.R. Neureuther, "SIMPL-2 (SIMulated Profiles from the Layout - Version 2)", Proceedings of the IEEE Symposium on VLSI Technology, Kobe, Japan, Digest of Technical Papers, pp. 64-65, May 1985.

[4] W.G. Oldham, A.R. Neureuther, C. Sung, J.L. Reynolds, and S.N. Nandgaonkar, "A General Simulator for VLSI Lithography and Etching Process: Part I - Application to Projection Lithography", IEEE Transactions on Electron Devices, vol ED-26, no 4, pp. 717-722, April 1979.

[5] C.P. Ho, J.D. Plummer, S.E. Hansen, and R.W. Dutton, "VLSI process modeling -- SUPREM-III", IEEE Transactions on Electron Devices, vol ED-30, no 11, pp. 1438-1453, November 1983.

[6] S.R. Nassif, A.J. Strojwas, and S.W. Director, "FABRICS II: A statistically based IC fabrication process simulator", IEEE Transactions on Computer-Aided Design, vol. CAD-3, pp. 20-46, January 1984.

[7] R.A. Hughes and J.D. Shott, "The Future of Automation for High-Volume Wafer Fabrication and ASIC Manufacturing", Proceedings of the IEEE, vol 74, no 12, pp. 1775-1793, December 1986.

[8] A.J. MacDonald, A.J Walton, J.M. Robertson, and R.J. Holwill, "Integrating CAM and Process Simulation to Enhance On-line Analysis and Control of IC Fabrication", IEEE Transactions on Semiconductor Manufacturing, vol 3, no 2, pp. 72-79, May 1990.

[9] K.K. Lin, and C.J. Spanos, "Statistical Equipment Modeling for VLSI Manufacturing: an Application for LPCVD", IEEE Transactions on Semiconductor Manufacturing, vol 3, no 4, pp. 216-229, November 1990.

[10] Z.M. Ling, S. Leang, and C.J. Spanos, "A Lithographic Workcell Monitoring and Modelling Scheme", Microcircuit Engineering Conference, Leuven, Belgium, September 1990.

[11] G.S. May, J.H. Huang, and C.J. Spanos, "Statistical Experimental Design in Plasma Etch Modeling," IEEE Transactions on Semiconductor Manufacturing, vol 4, no 2, May 1991.

[12] H.C. Wu, A.S. Wong, Y.L. Koh, E.W. Scheckler, A.R. Neureuther, "Simulated Profiles from the Layout, Design Interface in X (SIMPL-DIX)", IEDM Technical Digest, pp. 38-331, Dec. 1988.

[13] P. Sutardia, W.G. Oldham, "Modeling of Stress-Effects in Silicon Oxidation Including Non-Linear Ciscosity Oxide", IEDM Technical Digest, Washington D.C., pp. 264-267, Dec. 1987.

[14] K. Lee, Y. Sakai, A.R. Neureuther, "Topography-Dependent Electrical Parameter Simulation for VLSSI Design", IEEE Transactions on Electron Devices, vol ED-30, pp. 1469-1474, November 1983.

[15] K.K.H. Toh, A.R. Neureuther, "Identifying and Monitoring Effects of Lens Aberrations in Projection Printing", Proceedings of the SPIE -Optical Microlithography VI, vol 772, pp. 202-209, 1987.

[16] M.E. Law, C.S. Rafferty, R.W. Dutton, "SUPREM-IV Technical Report", Stanford Electronics Laboratory, Stanford University, 1988.

[17] M.R. Pinto, C.S. Rafferty, R.W. Dutton, "PISCES-II: Poisson and Continuity Equation Solver", Integrated Circuits Laboratory, Stanford University, September, 1984.

[18] A. Nye and T. O'Reilly, The Definitive Guides to the X Window System, 1st Edition, O'Reilly & Associates, Inc. January 1990.

[19] K.K. Lin, Private Communication, May 1991.

[20] S. G. Duvall, "An Interchange Format for Process and Device Simulation", IEEE Transactions on Computer-Aided Design, vol 7, no 7, pp. 741, July 1988.

[21] A.S. Wong and A.R. Neureuther, "The Intertool Profile Interchange Format - A Technology CAD Approach", to appear in the IEEE Transactions on Computer-Aided Design, September, 1991.

[22] L. A. Rowe, C. Williams and C. Hegarty, "The Design of the Berkeley Process-Flow Language", Electronics Research Laboratory report, University of California at Berkeley, August 1990.

[23] A. S. Wong et al, Minutes of the Standard Wafer Representation (SWR) meeting, March 1991.

Appendix A

A list of source code modifications in the SIMPL-IPX system

The following SIMPL-2 and SIMPL-DIX files have been modified to allow the access to the BCAM system:

SIMPL-2:

- Deposition.c
- Do_Process.c
- F77Layers.c

SIMPL-DIX:

- command.h
- default.h
- command_control.c
- dix_action2.c

Appendix B

Worst Case Analysis Interface for LPCVD

```
/*

Worst case analysis Interface
equipment model: LPCVD

Copyright (C) 1991 University of California BCAM Group.

Author: Tom L. Luan
Version: 5/1/1991
Source: /home/radon1/users/hodges/luan/forms
Reference: The Definitive Guides to the X Window System,
 vol 4&5. by Adrian Nye and Tim O'Reilly, 1990

 */


#include <stdio.h>

/*
 * Include files required for all Toolkit programs
 */
#include <X11/Intrinsic.h>/* Intrinsics Definitions */
#include <X11/StringDefs.h>/* Standard Name-String definitions */

/*
 * Public include file for widgets actually used in this file.
 */
#include <X11/XawMisc.h>
#ifdef X11R3
#include <X11/Form.h>
#include <X11/Command.h>
#include <X11/AsciiText.h>
#include <X11/Label.h>
#include <X11/List.h>
#else /* R4 or later */
#include <X11/Xaw/Form.h>
#include <X11/Xaw/Command.h>
#include <X11/Xaw/AsciiText.h>
#include <X11/Xaw/Label.h>
```

```
#include <X11/Xaw/List.h>
#endif /* X11R3 */

FILE *fp, *fp_w, *fopen();

/*
 * "Help" button callback function
 */

void Help(w, topLevel, call_data)
Widget w;
Widget topLevel;
caddr_t call_data;
{
 system("xtroff -full help.x &");
}

 /*
 * combination of thickness and thickness uniformity
 */
void AXB(w, client_data, call_data)
Widget w;
caddr_t client_data, call_data;
{
float thickness, t_sigma, thick_uniformity, t_u_sigma;
float k, nominal, low, high, nonmax;

fp = fopen ("stat.in", "r");
 fscanf(fp, "%g,%g,\n%g,%g\n", &thickness, &t_sigma, &thick_uniformity, &t_u_sigma);

k = 1.96; /* the constant before sigma, for 95% confidence level */
nonmax = 1 - (thick_uniformity - k * t_u_sigma); /* max non-uniformity */

nominal = thickness; /* nominal thickness */
low = thickness - k * t_sigma - thickness * nonmax; /* low value */
high = thickness + k * t_sigma + thickness * nonmax; /* high value */

/*
 * to show on the screen:
 printf("\n%g\n%g\n%g\n\n", nominal, low, high);
 *
 */

/* write nominal, low, and high values into a file to be read
```

```
by SIMPL-IPX program
*/

fp_w = fopen ("stat.out", "w");
fprintf(fp_w, "%g\n%g\n%g\n", nominal, low, high);
fclose(fp_w);

fclose(fp);

exit(0);
}

/*
 * "Quit" button callback function
 */
void Quit(w, client_data, call_data)
Widget w;
caddr_t client_data, call_data;
{
exit(0);
}

/*
 * Main function
 */
main(argc, argv)
int argc;
char *argv[];
{
 Widget topLevel, box, uti, utiName, proc, procDes, pram1, pram1Def,pram2, pram2Def, pram3, pram3Def,
pram4, pram4Def, help, aXb, aXbXc, aXbXcXd, quit;

 if(argc != 2)
printf("usage: %s <filename> [&]\n", argv[0]);

else{ /* set up the widgets */

topLevel = XtInitialize(
argv[0],/* Application name */
argv[1],/* Application class */
NULL,/* Resource Mgr. options */
0,/* number of RM options */
&argc,/* number of args */
argv/* command line */
```

```
);

box = XtCreateManagedWidget(
"box", /* arbitrary widget name */
formWidgetClass,/* widget class from Box.h */
topLevel,/* parent widget*/
NULL,/* argument list */
0/* arg list size */
);

/*
 * Set up the equipment and equipment name widgets.
 */
uti = XtVaCreateManagedWidget(
"uti", /* arbitrary widget name */
labelWidgetClass, /* widget class from Label.h */
box, /* parent widget*/
XtNlabel, "Utility:",
XtNborderWidth, 0,
NULL
);

utiName = XtVaCreateManagedWidget(
"utiName", /* arbitrary widget name */
labelWidgetClass, /* widget class from AsciiText.h */
box, /* parent widget*/
XtNjustify, XtJustifyLeft,
XtNborderWidth, 0,
XtNborderWidth, 0,
XtNfromHoriz, uti,
NULL
);

/*
 * Set up the process and process description widgets.
 */
proc = XtVaCreateManagedWidget(
"proc", /* arbitrary widget name */
labelWidgetClass, /* widget class from Label.h */
box, /* parent widget*/
XtNlabel, "Definitions:",
XtNborderWidth, 0,
XtNfromVert, uti,
NULL
```

```
);

procDes = XtVaCreateManagedWidget(
 "procDes", /* arbitrary widget name */
 labelWidgetClass, /* widget class from AsciiText.h */
 box, /* parent widget*/
XtNjustify, XtJustifyLeft,
 XtNborderWidth, 0,
XtNfromVert, uti,
XtNfromHoriz, proc,
 NULL
 );

/* parameter definitions */
 pram1 = XtVaCreateManagedWidget(
 "pram1", /* arbitrary widget name */
 labelWidgetClass, /* widget class from Label.h */
 box, /* parent widget*/
 XtNlabel, "T = ",
 XtNborderWidth, 0,
 XtNfromVert, proc,
 NULL
 );

pram1Def = XtVaCreateManagedWidget(
 "pram1Def", /* arbitrary widget name */
 labelWidgetClass, /* widget class from AsciiText.h */
 box, /* parent widget*/
XtNjustify, XtJustifyLeft,
 XtNborderWidth, 0,
XtNfromVert, proc,
XtNfromHoriz, pram1,
 NULL
 );

 pram2 = XtVaCreateManagedWidget(
 "pram2", /* arbitrary widget name */
 labelWidgetClass, /* widget class from Label.h */
 box, /* parent widget*/
 XtNlabel, "TU = ",
 XtNborderWidth, 0,
 XtNfromVert, pram1,
 NULL
 );
```

```
pram2Def = XtVaCreateManagedWidget(
 "pram2Def", /* arbitrary widget name */
 labelWidgetClass, /* widget class from AsciiText.h */
 box, /* parent widget*/
XtNjustify, XtJustifyLeft,
 XtNborderWidth, 0,
XtNfromVert, pram1,
XtNfromHoriz, pram2,
 NULL
 );

 pram3 = XtVaCreateManagedWidget(
 "pram3", /* arbitrary widget name */
 labelWidgetClass, /* widget class from Label.h */
 box, /* parent widget*/
 XtNlabel, "S = ",
 XtNborderWidth, 0,
 XtNfromVert, pram2,
 NULL
 );

pram3Def = XtVaCreateManagedWidget(
 "pram3Def", /* arbitrary widget name */
 labelWidgetClass, /* widget class from AsciiText.h */
 box, /* parent widget*/
XtNjustify, XtJustifyLeft,
 XtNborderWidth, 0,
XtNfromVert, pram2,
XtNfromHoriz, pram3,
 NULL
 );

 pram4 = XtVaCreateManagedWidget(
 "pram4", /* arbitrary widget name */
 labelWidgetClass, /* widget class from Label.h */
 box, /* parent widget*/
 XtNlabel, "SU = ",
 XtNborderWidth, 0,
 XtNfromVert, pram3,
 NULL
 );

pram4Def = XtVaCreateManagedWidget(
```

```
"pram4Def", /* arbitrary widget name */
labelWidgetClass, /* widget class from AsciiText.h */
box, /* parent widget*/
XtNjustify, XtJustifyLeft,
XtNborderWidth, 0,
XtNfromVert, pram3,
XtNfromHoriz, pram4,
NULL
);


/*
 * Set up the command widgets
 */

aXb = XtVaCreateManagedWidget(
"aXb",
commandWidgetClass,
box,
XtNlabel, "T & TU ",
XtNfromVert, pram4,
NULL
);

aXbXc = XtVaCreateManagedWidget(
"aXbXc",
commandWidgetClass,
box,
XtNlabel, "T & TU & S",
XtNfromVert, aXb,
NULL
);

aXbXcXd = XtVaCreateManagedWidget(
"aXbXcXd",
commandWidgetClass,
box,
XtNlabel, "T & TU & S & SU",
XtNfromVert, aXbXc,
NULL
);

help = XtVaCreateManagedWidget(
"help",
```

```
commandWidgetClass,
 box,
 XtNlabel, "HELP",
 XtNfromVert, aXbXcXd,
 NULL
 );

quit = XtVaCreateManagedWidget(
"quit",
commandWidgetClass,
 box,
 XtNlabel, "QUIT",
 XtNfromVert, aXbXcXd,
 XtNfromHoriz, help,
 NULL
 );

/*
 * List of callbacks
 */
XtAddCallback(aXb, XtNcallback, AXB, 0);
XtAddCallback(quit, XtNcallback, Quit, 0);
XtAddCallback(help, XtNcallback, Help, topLevel);

/*
 * Create windows for windows for widgets and map them
 */
XtRealizeWidget(topLevel);

/*
 * Loop for events
 */
XtMainLoop();
}
}
```

```
**************************************
* Resource Settings for the above Program *
**************************************
*background: lightgray
*utiName.label: CHOOSE WORST CASE COMBINATION
*procDes.label:
*pram1Def.label: Thickness
*pram2Def.label: Thickness Uniformity
*pram3Def.label: Built-in Stress
*pram4Def.label: Stress Uniformity
*pram1Def.font: *courier-bold-o-*-120*
*pram2Def.font: *courier-bold-o-*-120*
*pram3Def.font: *courier-bold-o-*-120*
*pram4Def.font: *courier-bold-o-*-120*

*aXb.width: 145
*aXb.height: 25
*aXbXc.width: 145
*aXbXc.height: 25
*aXbXcXd.width: 145
*aXbXcXd.height: 25
*help.width: 60
*help.height: 40
*quit.width: 60
*quit.height: 40

*Command.background: lightblue
*help.foreground: white
*help.background: blue
*quit.foreground: white
*quit.background: blue
```