

Chapter 2. Computer-Integrated Design and Manufacture of Integrated Circuits

Academic and Research Staff

Professor Donald E. Troxel, Professor Dimitri A. Antoniadis, Professor Duane S. Boning, Linus F. Cordes, Gregory T. Fischer, Stanley B. Gershwin, Thomas J. Lohman, Michael B. McIlrath

Graduate Students

Asbjorn M. Bonvik, John C. Carney, Yonald Chery, Jarvis B. Jacobs, James T. Kao, Somsak Kittipiyakul, Jimmy Y. Kwon, William P. Moyne, Joseph E. Nemece, Nadir E. Rahman, Chantal E. Wright

Technical and Support Staff

Francis M. Doughty

2.1 Research Goals and Objectives

Sponsors

Defense Advanced Research Projects Agency/
U.S. Navy
Contract N00174-93-K-0035
Defense Advanced Research Projects Agency/
U.S. Army
Contract DABT 63-95-C-0088
Multisponsored Projects Industrial/
MIT Leaders for Manufacturing Program

2.1.1 The MIT Computer-Aided Fabrication Environment

The computer-aided fabrication environment (CAFE) is a software system being developed at MIT for use in the fabrication of integrated circuits and microstructures. The distinguishing feature of CAFE is that it can be used in all phases of process design, development, planning, and manufacturing of integrated circuit wafers. CAFE presently provides day-to-day support to research and production facilities at MIT with both flexible and standard product capabilities. This manufacturing software system is unique in the development of a process flow representation and its integration into actual fabrication operations. CAFE provides a platform for work in several active research areas, including technology (process and device) computer-aided design (TCAD), process modeling, manufacturing quality control, and scheduling.

Architecture

The CAFE architecture is a computer integrated manufacturing (CIM) framework for the deployment and integration of integrated circuit (IC) and process design and manufacturing software. CAFE uses an object oriented database model which is implemented in a layered manner on top of a relational database. Our database schema is based on GESTALT, an object oriented, extensible data model. GESTALT is a layer of abstraction which maps user defined objects onto existing database systems (e.g., a relational DBMS) and shields application programs from the details of the underlying database. The integration architecture includes the conceptual schema and models used to represent the IC manufacturing domain in CAFE and the user and programmatic interfaces to the various applications. Two important CAFE applications relate process simulation and actual wafer fabrication to the same process flow representation.

CAFE Applications

The fabrication of wafers with a process represented as a process flow representation (PFR) involves several steps. A suitable PFR for the specific process must be created and installed. Wafer lots must be created and associated with this specific PFR. These lots must then be "started" to create a task data structure which is isomorphic to the hierarchical structure of the PFR.

At this point, actual machine operations can be scheduled and reservations made for both machines and operators. Finally, the machine operations can be performed, instructions given to the operator and machines, and data collected from the operator or machine and entered into the database.

2.1.2 National Infrastructure for Networked Design and Prototyping

We are developing tools and infrastructure to enable virtual and physical prototyping of advanced microsystems. A "quasi-empirical" simulation approach requires data from select experiments to build accurate performance, reliability, and variation or statistical models. Parameters for these models must be determined and refined by multiple series of both simulations and repeated experiments. The validation of simulations produced by the computational prototyping tools can only be achieved by correlation of simulated results with actual experimental measurements. Advanced computational prototyping tools and methodologies will enable collaborative networked subsystem design and physical prototyping.

The infrastructure development will enable management of information required to specify processing, collect measurements, and retrieve results of remote processing and metrology. This infrastructure will support the timely conduct of the multitude of experiments to calibrate and validate simulations which are embodied in the computational prototyping of high performance integrated circuits.

Models and tools developed under this program, as well as both simulated and experimental data, will be collected into libraries and repositories and made available via network. This will provide convenient and effective access to research results as they become available and support collaborative subsystem design and physical prototyping.

2.1.3 CAFE at MIT Lincoln Laboratory

Project Staff

Gregory T. Fischer, Thomas J. Lohman, Professor Donald E. Troxel

MIT Lincoln Laboratory is using CAFE as the CIM system in their IC processing facility. Multiple lots are being processed daily via process flow representation (PFR) based fabrication. Lincoln has 100 percent of their wafers in this facility processed with PFR based fabrication.

It is now possible to view CAFE Help and Lab Manual text via the World Wide Web. Options have been added under the "Help" and "Help"/"Equipment Help" menus which will initiate a web browser to display the specified help text section.

2.2 Operating High Variability Manufacturing Systems

Project Staff

Asbjorn M. Bonvik

We present two factory control policies derived from optimal control considerations. Using sample factories distributed by Sematech, we compare the performance of these policies to other control strategies such as kanban and periodic lot release. We show that the policies designed for high variability environments have very attractive properties, including higher throughput than periodic release and much lower inventories than kanban.

Semiconductor manufacturing differs from many other manufacturing environments by the complexity and variability of its processes. A semiconductor process can be several hundred operations long, and the machines are relatively unreliable due to the leading edge technology involved. This makes semiconductor factories harder to control than other plants of comparable size.

We construct control policies by extending the structural properties found by optimal control formulations of smaller systems. This allows us to construct good, although not provably optimal, control schemes for large production systems.

In the following, we explain the structure of two such policies and contrast them with control policies that are commonly found in industry. We then report simulation results from datasets distributed by Sematech, representing actual semiconductor factories.

We have built a scheduling and simulation testbed integrated with the MIT CAFE computer-aided fabrication environment. This program can act as a factory floor dispatcher, where it recommends a daily schedule based on the factory state stored in the database. The program can also run in a simulation mode, where it will project long-term consequences of following a certain scheduling policy under various assumptions on lot releases and machine reliabilities. The main architectural components are (1) the data structures that represent the system state, (2) the front ends that create the initial state, (3) the system dynamics module that makes the state evolve in time, and (4) the control modules that make release and dispatch decisions.

In its simulation mode, the testbed can read the Sematech datasets instead of the CAFE database. This allows us to test out our scheduling approaches in other manufacturing systems than our own. The testbed does not currently support all

features found in the Sematech datasets. For instance, we do not simulate operator availability, scrap, or rework. Setups are simulated, but the scheduling policies tested do not make any attempt to optimize the setup sequence.

This leads to different production rates from our testbed than from the Delphi simulator also distributed by Sematech. On the other hand, we allow the scheduling modules to control the lot release rates to the fabrication, not just manipulate the dispatch decisions as appears to be the case with Delphi. Previous work has shown that the lot releases are a more important factor in fabrication performance than lot dispatch decisions on the shop floor.

2.3 A Distributed Discrete Event Simulation

Project Staff

Asbjorn M. Bonvik

Discrete event simulation is, almost by definition, a very CPU-intensive task. Some stochastic processes are represented by a program incorporating a pseudo-random number generator, and statistics are collected by executing the program. It may be necessary to run the simulation for a very long time to get good estimates of the performance measures of the actual system under study. For example, if one is studying a manufacturing system where machine failures are important to the overall behavior of the system, the simulation needs to cover a period with many failures of different lengths.

It is very hard to parallelize a discrete event simulation, because the commonly found algorithm maintains a centralized event list and a current simulation time, executing one event at a time. The current event may manipulate the list of future events, for instance by scheduling the end of some activity or the next instance of some event. When the current event is done, the event with the lowest activation time is taken out of the event queue, the simulation time is set to this value, and the event is executed. This essentially presumes a single thread of execution on a single processor.

On the other hand, because of statistical considerations, it is common to split a large simulation job into a batch of independent replications, where each replication is started from a different random number seed, everything else being the same. This is done to eliminate the dependency on a single sequence of pseudo-random numbers and to establish confidence intervals on the performance measures. This also gives a natural granularity for a

coarse-grained parallel or distributed computation: Distribute one replication to each processor, and designate one processor to collect the statistics emanating from the computation.

This is exactly what we did for a recent study of a small production system under various conditions. Our work had two phases: a screening phase where a large number of parameter settings had to be evaluated, and a refinement phase in which multiple replications were done of the best parameter settings to draw final conclusions. To do this efficiently, we constructed a distributed simulator architecture that ran on about one dozen Sun SparcStations, mainly SparcStation 10s. The computers used were the MTL machines, the machines at the Operations Research Center, and a single SparcStation 10 at the Norwegian Defence Research Establishment. After the actual study was completed, a Pentium 100 MHz PC running Linux was added to the architecture. This demonstrates that the architecture is not Sun-specific. Incidentally, the PC turned out to work approximately 20 percent faster than even a SparcStation 20, possibly because of faster I/O leading to less overhead when starting a client.

The software had three components: the discrete event simulation program itself, a client program that drove a single replication on some host, and the server program that organized the computation and collected statistics.

By using these programs, we successfully simulated about 6000 different cases. Most cases were run for a single replication of about 100 weeks of simulated time, but about 50 cases were selected for further scrutiny. This required 50 more replications of each case, for a total of about 8500 replications. We estimated this to take about three months of CPU time on a single SparcStation 2 such as hierarchy. Using our virtual supercomputer, the entire job was done in four days of real time. In addition, we did extensive model validation runs and a number of false starts before getting it right.

More speed is available by using more hosts. The approach is not limited to Sun computers, as our Pentium PC demonstrated. Anything that supports UNIX stream sockets can be used. This allows other fast computers to be added by the simple expedient of getting an account, recompiling the simulation for that machine type, and adding the host name to the server list of target hosts.

In conclusion, our approach gives supercomputer speed at very little cost. Suitable jobs for a similar scheme are very CPU-intensive with small memory and I/O requirements and have a structure such that fairly large pieces of the job can be computed

separately and the results assembled with little effort.

2.4 QUAN: A Language for the Schedule Of Repetitive Manufacturing Systems

Project Staff

Joseph E. Nemeč

We propose the syntax for a language called Quan which allows for the expression of scheduling policies for repetitive manufacturing systems—systems which make multiples of single or several part types. We first develop a representation of the factory using an object-oriented approach similar to that of the one found in Stamatopoulos.¹

In the course of doing so, we describe the physical objects in the factory, including such objects as machines, buffers, parts, and technicians, as well as the conceptual objects in a factory, including manufacturing cells, process flows, operations, process constraints, and material constraints.

2.4.1 Introduction

We propose a language that will allow for the expression of scheduling policies for repetitive manufacturing systems. We propose to use an object-oriented approach, using the software developed by Stamatopoulos¹ as a basis.

In order to describe scheduling policies in the scheduler in a consistent manner, we must be able to express what constitutes the physical objects in the factory, such as machines, buffers, technicians, parts, as well as how these objects interact with each other, what their characteristics are, and other attributes. To this end, we define several representations; the factory representation (FR), the material representation (MR), the process representation (PR), the scheduling representation (SR), and the constraint representation (CR). Each will be described in detail.

In addition to the representations, we also have the state of the system. This is represented as the factory state (FS) and the materials state (MS).

With the complete definition of the factory at our disposal, we will then describe the quantity scheduling language (Quan), which will be interpreted by the scheduler. This language will allow for the expression of inventory control policies, such as Kanban or CONWIP,² or hybrids of such policies³ and production policies, such as FIFO, LIFO or preemptive processing.⁴

2.4.2 The General Model

Figure 1 shows the interaction between the various representations, the various states and the scheduler.

All of the representations, except the SR, are static. Once defined and internalized, they are not consulted anymore. They form the backbone of the manufacturing system. For example, it is assumed that once it has been defined that a certain operation can only be performed on a certain machine with certain operators, these constraints will not change. If, for example, more workers were trained to work a specific machine, then the representation of the factory must be updated to take this change into account. As such things do not happen instantaneously, but rather require time and effort, they are not of immediate concern to the scheduler.

The SR is continuously consulted by the scheduler. The scheduler, in making decisions, may have several possibilities for each machine. For example, we may make it a point of always running with a First-In First-Out unless we are badly backlogged, in which case we begin prioritizing certain part-types over others. This change in the factory state (i.e. the backlog) necessitates a reconsultation with the SR to determine appropriate actions.

The consultative process of the scheduler is illustrated in figure 2.

¹ M. Stamatopoulos, *A Factory Representation as a Design Tool in a Computer Integrated Manufacturing Environment*, M.S. thesis, MIT, 1994.

² J.A. Buzacott, J. Shanthikumar, and J. George, *Stochastic Models of Manufacturing Systems*, (Englewood Cliffs, New Jersey: Prentice-Hall, 1993).

³ A.M. Bonvik, Y. Dallery, and S.B. Gershwin, "Approximate Analysis of Production Systems Operated by a CONWIP-finite Buffer Hybrid Control Policy," submitted for publication.

⁴ L. Kleinrock, *Queueing Systems, Volume 1: Theory* (New York: Wiley-Interscience, 1975).

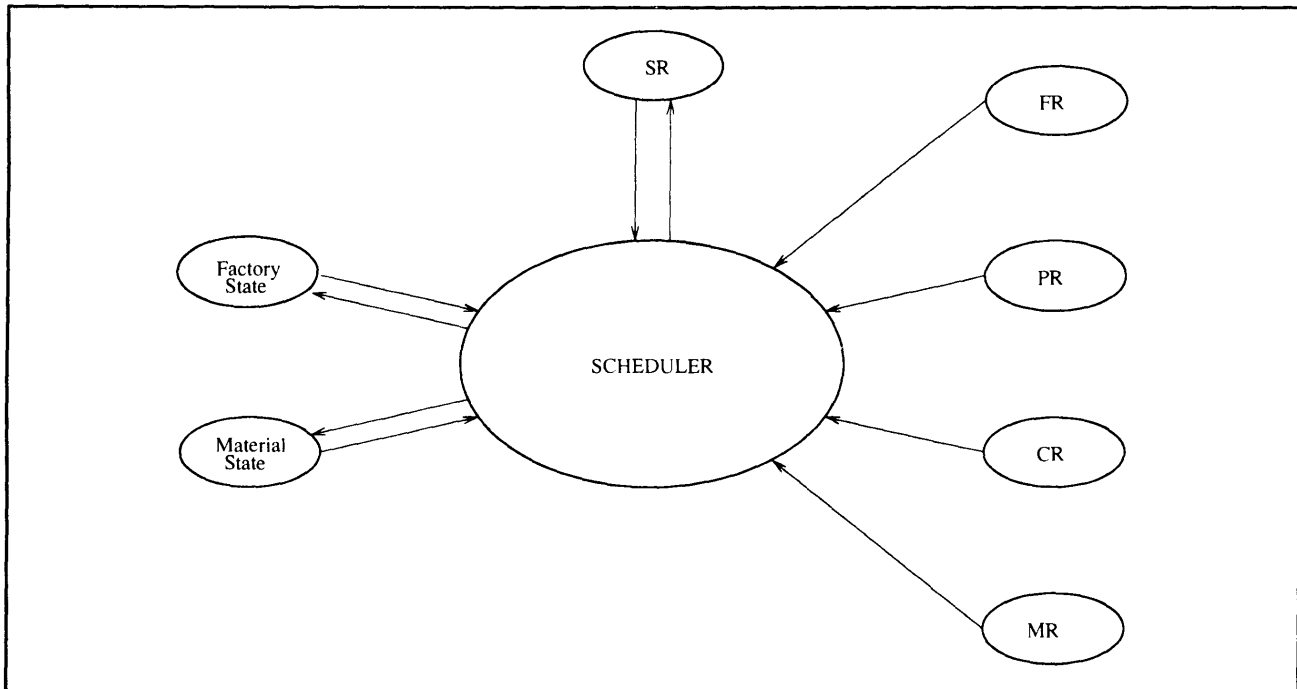


Figure 1.

2.5 Microsystems Factory Representation

Project Staff

Joseph E. Nemec

The microsystems factory representation (MFR) is a system that allows for the explicit representation of factories using an object oriented, C-like block structure programming language. It allows for the easy declaration of the attributes of various objects, and for fast and powerful editing of specific instances of the attributes defined. This document will describe the MFR through several examples, demonstrating its flexibility and power.

The MFR has three fundamental parts: the Schematic Representation, the Textual Representation, and the Textual Interpreter. The procedure for representing a factory in the MFR comes in two stages. In the first stage, the objects that are in the factory are declared. These include such classes of physical objects as machines, buffers, technicians, etc., as well as artificial constructions dependent on the structure of the factory chosen, such as cells. This collection of objects is known as the schema of the representation. This schema is compiled into the MFR Textual Interpreter by the MFR Schematic Interpreter.

In the second stage, the MFR textual representation of the factory, the representation of how the information about the factory is organized, is specified.

The syntax of the textual representation is dependent on the schema declared. The MFR Textual Interpreter is then used to parse the text, check for syntax errors, and create a representation of the factory in computer memory, or instantiate it into a data base.

The MFR schema is simply the collection of objects of different classes required to create an MFR textual representation of the factory. This is where the characteristics of such objects as machines or buffers are defined. For example, if the object we are creating is a machine class, we will want the class to have such attributes as mean time to fail and mean time to repair, failure modes, etc. The syntax of the schema representation is block-structured, like C. A programmer's manual for the schema may be found in Stamatopoulos.¹ The schematic is then compiled into the MFR Textual Interpreter using the MFR Schematic Interpreter.

The MFR textual representation describes how the objects defined in the schema are organized in the factory. Specifically, it declares how many of each object there are, their names, their locations, their specific attributes, etc. The textual representation allows for a hierarchical representation of the objects or a one-dimensional representation. We will see later that a hierarchical representation is preferred.

The MFR textual representation is then read by the MFR Textual Interpreter, which parses the text, performs consistency checks on the data and, creates

a representation of the MFR objects (those defined by the MFR schema above) in computer memory.

2.6 Run-by-Run: Interfaces, Implementation, and Integration

Project Staff

William P. Moyne

Run-by-run (RbR) control is a form of adaptive model based process control where recipe changes are performed between runs of the process. It is becoming popular in the area of VLSI processing but its acceptance has been hindered by integration issues. Existing systems cannot be replaced due to massive capital investments, so the RbR controller

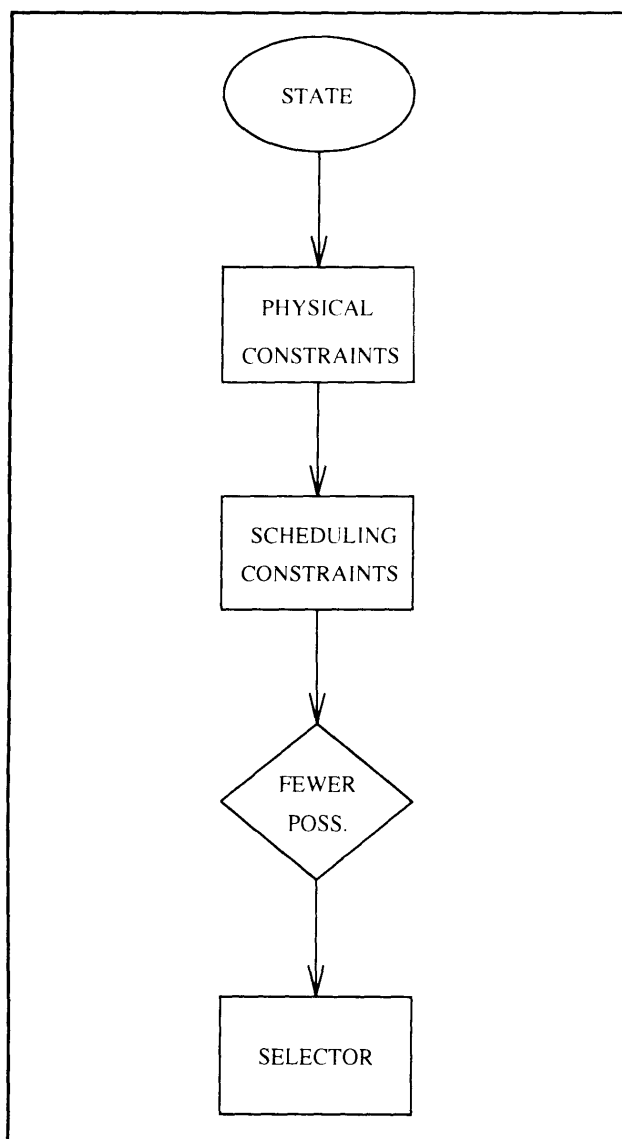


Figure 2.

must mesh with these systems without requiring large modifications. Two steps have been taken to ease this integration. First, an abstract data model has been developed for RbR control which can be easily communicated between dissimilar modules. Second, a three tier communication model has been developed to allow multiple levels of interaction with the RbR control module. These layers complement the underlying data model.

An RbR control server has been implemented to demonstrate the robustness of the communication model and data abstraction. This server provides RbR control to a variety of clients via TCP/IP network sockets. One of these clients is a graphical user interface that allows direct operation of the control algorithms. This can be a powerful tool when evaluating new control algorithms or testing equipment models. The interface contains a set of simulation, graphing, and archiving tools to aid in the testing and operation of the server. The controller has been integrated into a local computer integrated manufacturing (CIM) system, as well as a control framework being developed by The University of Michigan and SEMATECH.

In addition to interface issues, the control algorithms themselves have been enhanced to enable a variety of constraints and bias terms to be added to the models. The controller currently contains two control algorithms, but is designed to be expanded to include more algorithms as they are developed. Data needed for these new algorithms can be constructed using the data abstraction without disrupting existing modules. Implementation, interface, and integration barriers to the adoption of run-by-run control have been reduced by the definitions and demonstrations presented in this thesis.

2.7 Remote Fabrication of Integrated Circuits

Project Staff

Jimmy Y. Kwon

The computer-aided fabrication environment (CAFE) is a software system developed at MIT for use in all phases of integrated circuit fabrication. While still undergoing development and enhancements, CAFE provides day-to-day support to research and production facilities at MIT, with both standard and flexible product lines.

One of the limitations of the present CAFE system is that it is not a fully open system. An open system is one that is designed to accommodate components from various software applications and can be viewed from three perspectives: portability, inte-

gration, and interoperability. This document is concerned with the idea of interoperability between different computer integrated manufacturing (CIM) systems, and describes the extension to CAFE's architecture to support what is called remote fabrication.

With the goal of general interoperability between different CIM systems in mind, the virtual object manager is developed. The virtual object manager provides a framework for three key modules to support interoperability: the Object Management Module, the Export/Import Module, and the Remote Message Passing Module. Each of these modules are discussed and implemented for the CAFE platform.

Building upon the interoperability tools in the virtual object manager, application layer support is then developed for the specific task of remote fabrication in CAFE. The application modules and additional software tools are described, and together form the CAFE remote fabrication system, a prototype system providing remote fabrication capabilities. An example of a remote fabrication session is described, stepping the reader through CAFE's remote processing paradigm and showing how the various tools are used. In the form of a tutorial, the example starts with installing a process flow into the CAFE database and creating a lot of wafers. The operate-machine/next-operation processing cycle is then done using both local and remote machines, showing the interoperability features at work. When the processing cycle completes a traveller file is generated, summarizing the entire history of processing.

While this document describes and implements a remote fabrication system specifically for CAFE, it is hoped that this document brings into focus some of the issues involved in general interoperability between different CIM systems and provides a useful framework for future work.

2.8 Message Passing Tools for Software Integration

Project Staff

John C. Carney

As the base of developed software grows, software integration is becoming increasingly important. Many software systems are large and complex. Since rewriting the entire system is not generally possible, newly developed programs must be integrated into the existing system. It is the responsibility of the developer to not only write a new

program or tool, but to make it work within the existing environment.

There are several approaches to software integration. Programs within the Unix environment typically are integrated by operating on a common set of files. One program is used to create a file, another to process it, and perhaps yet another to analyze the results. Other approaches include program databases and remote procedure calls (RPCs). An alternative approach for software integration is through the use of a message passing system. The goal of the message passing system is to provide a method by which structured information may be exchanged between two or more running processes. These processes may possibly be running on a single workstation or on multiple workstations connected by a network.

While there are not many message passing systems in existence, the majority which do exist have been designed to be used for a special purpose or in a particular context. There are some general purpose message passing systems, however, these systems impose a fixed integration architecture and message format. A more flexible peer-to-peer approach to message passing has been developed. The peer-to-peer messaging system allows the software developer and integrator to completely design and choose the desired integration architectures and message formats.

The communications handling application tool suite (CHAT) has been developed and provides the programmer with a suite of libraries which can be used to integrate software programs using the peer-to-peer message passing approach. There are three libraries of routines, providing support for packaging data into messages, handling incoming messages, and application connection. All three libraries are implemented both in C and Tcl/Tk, a scripting language and toolkit for creating graphical user interfaces under X-Windows. The CHAT suite will allow arbitrary combinations and architectures of C programs and Tcl/Tk programs to exchange messages.

Libraries within the CHAT suite have been used to integrate a new factory display program into CAFE, MIT's computer-aided fabrication environment. CAFE is a software system for the use in the manufacture of integrated circuits, and provides day-to-day support for both research and production facilities at MIT. The factory display program is a graphical program which displays a map of a semiconductor manufacturing facility, including the machines and lots within the facility. The CHAT suite has also been used within other software integration efforts and plans have been made to use the libraries in future integration projects within

MIT's Computer Integrated Design and Manufacturing (CIDM) project.

2.9 Remote Microscope for Inspection of Integrated Circuits

Project Staff

James T. Kao

The remote microscope was developed at MIT as part of the computer integrated design and manufacturing project to aid in the remote fabrication of integrated circuits, and will allow a user to operate and view in "real time" an actual microscope located at a distant facility. We envision a growing trend in the semiconductor processing field for more collaboration and sharing of resources, so it will become important for researchers to have access to a telemicroscopy system like the one developed at MIT to perform remote inspections of semiconductor wafers.

The MIT remote microscope is extremely versatile; it operates over the internet and allows a user to run the graphical microscope interface on any ordinary UNIX workstation, thereby providing easy access to the microscope for researchers located throughout the world. The actual remote microscope utilizes readily available hardware as well, making the entire system very straightforward and economical to implement. To the best of our knowledge, the MIT remote microscope is the first telemicroscopy system to operate on the internet. The remote microscope also is original because it allows multiple users to view the microscope at the same time in a conference inspection, which should be very useful when geographically distant researchers wish to collaboratively inspect a wafer. The overwhelming response to the remote microscope has been very positive, as it was officially demonstrated in August 1995 between MIT and Stanford University during the annual TCAD symposium.

2.10 Semiconductor Manufacturing Process Flow Representation (PFR)

Project Staff

Michael B. McIlrath

2.10.1 Introduction

The process flow representation (PFR) and its integration into both design and fabrication operations is central to the CIDM program and the computer-aided fabrication environment (CAFE) software system.³ CAFE is currently being used at the semiconductor fabrication processing facilities of the MIT Microsystems Technology Laboratories, Lincoln Laboratories, and Case Western University. Activities in this area include both formal methods for process modeling and the practical application of process representation to process design and execution.

For high performance computing systems and other advanced technology, concurrence in the design of the product, manufacturing process, and factory is crucial. The goal is to achieve fully integrated design and manufacture, in which the boundary between design and manufacturing domains is eliminated: in particular, information from the manufacturing floor is continuously available from the earliest stages of process and device design onward. Conversely, the manufacturing process, developed concurrently with the product, continues to undergo design via improvement and modification while in production. Computer integrated design and manufacture (CIDM), therefore, requires a coherent manufacturing process representation capable of storing information from a variety of different knowledge domains and disciplines and supporting access to this information in a consistent manner. We believe that our general semiconductor process modeling framework organizes the complexity of this interrelated information and puts our process flow representation (PFR) on a sound footing by giving it clear semantics.

A high level conceptual model for describing and understanding semiconductor manufacturing processing is a crucial element of both the CIDM research program, and of software frameworks for TCAD and CIM, including the MIT computer-aided fabrication environment (CAFE). Initially a two-stage generic process step model was used, which described processing steps in terms of two independent components: an equipment-dependent, wafer-independent stage, which maps equipment settings to physical processing environments, and an equipment-independent, wafer-dependent stage, which relates physical environments to changes in the input wafers. Driven by the needs of process control and optimization research, including sophisticated modeling, design, and experimental model verification, our fundamental conceptual process model has evolved from the two-stage generic process model into one which is part of a more general process modeling framework, in which the

earlier two-stage model is a special case. Our approach to process representation for both TCAD and CIM is based on this general modeling framework for semiconductor processing. In this framework, state information (e.g., wafer, environment, and equipment state), and models, or transformations, that describe relationships between state descriptions, are formally identified and described. The purpose of this comprehensive framework is to enable an effective representation that can be used throughout the IC semiconductor process life-cycle, from early conception and design phases through fabrication and maintenance.

In the MIT CAFE system, the PFR is expressed in a textual (ASCII) format and then converted into Gestalt objects and loaded into the CAFE database. The textual language of the PFR is extensible, so that it can flexibly accommodate changes and extensions to both the underlying modeling methodology and the needs of specific applications. The object-oriented nature of the Gestalt database interface enables the convenient evolutionary development of CAFE software applications built around the PFR. The PFR allows process step descriptions to be "underdetermined"; for example, by expressing only the wafer-state change, making it possible to develop a process incrementally with increasing degrees of detail. In addition to expressing the fundamental concepts of wafer transformation within individual process steps, the PFR supports both hierarchical and parameter abstraction and embedded computation, thereby providing support for modular process design and development. Processes expressed in the PFR can be simulated using a variety of technology CAD tools; PFR extensibility allows the incorporation of both simulator-dependent and simulator-independent information. A simulation manager application uses the appropriate information in the PFR along with knowledge of specific simulators to invoke simulation tools and maintain simulation state.

2.10.2 Process Development and Execution

The PFR has been used to develop and execute all of the fabrication processing at MIT Lincoln Laboratory during the last three years. Over 500 fabrication runs have been completed with around forty lots in process at any given time. These lots represent a variety of technologies, including CCD, low power CMOS, and SOI. All the Lincoln facility processing uses CAFE and the CAFE PFR.

Additionally, at MIT the PFR is used for both baseline and research processing; in particular, in support of other computer integrated design and

manufacturing (CIDM) and MTL research activities including hot carrier reliability and extreme submicron technology design.

2.10.3 Design Rule Checking for Wafer Fabrication Using CAFE

We have developed an extensible and general framework for process design rule checking. A table-driven approach used is to check for rule constraints at each processing step, based on the current state of the wafers. By "wafer state" we refer not only to the simulated or measured process effects but also to the process history. The rule-checker can be used in two ways: on-line, to prevent unsafe operations from occurring during fabrication and during process.

2.11 Computer-Aided Technology Design

Project Staff

Professor Dimitri A. Antoniadis, Professor Duane S. Boning, Michael B. McIlrath, Nadir E. Rahman, Jarvis B. Jacobs

2.11.1 Introduction

The design of advanced integrated circuit microsystems is increasingly tightly linked to the design of both the component devices of the microsystem and the microfabrication process used in its manufacture. Traditionally, the devices and manufacturing process used in building integrated circuits are collectively called the technology. The objective of microsystems technology design is to devise a fabrication process sequence which yields structures with some desired characteristics. Actually, two design activities proceed in parallel: design of the device structures and the process to fabricate them. In general, design space for both is explored by the technology designer using a combination of physical experiments and numerical simulation.

The microsystem (circuit) designer typically views the technology through a very limited interface; usually, a set of process parameters, which describe electrical behavior of underlying structures (for example, resistance of polysilicon), and the design rules, which express the allowable manufacturing limits in geometrical terms (i.e., minimum line width). The high-level goal of this project area is the development of tools and methodologies for more fully integrating the technology design with

both the product design and the manufacture of integrated circuit microsystems.

2.11.2 Technology CAD Framework

Contemporary large-scale software system engineering emphasizes frameworks, wherein common structure and interface specifications enable both current and future software components to be integrated in a flexible and modular way. Frameworks have been particularly successful in the development of electronic circuit CAD systems. Software may be roughly divided into tools, such as a simulator, which perform some part of an application task, and services, such as a database, which provide some necessary support capability used by various tools.

With framework standards, reusable, interchangeable software components from various suppliers may be deployed in systems which comply with the standard. In the broad sense, a framework standard specifies:

1. data representations for the objects of discourse in the application domain and their semantics, and programmatic interfaces to those representations, and
2. architecture; that is, interactions among software components (tools and services), and how tools fit together to perform application tasks for the user.

Standards for CAD frameworks are currently being established by the CAD Framework Initiative (CFI), a broad organization of vendor and user companies which has expanded its scope to include technology CAD. Technology CAD (TCAD) framework components include programming representations for the fundamental objects of process and device CAD: the physical structures on the wafer, the manufacturing process, and the structure and behavior of the resulting devices. A TCAD framework standard should also specify how application software is structured to use these representations and the underlying software services in process and device design and simulation activities.

The problem of wafer representation can be divided into geometry (shapes of regions and their relations) and fields (variations of properties over a region). A wafer representation for two-dimensional simulation has been designed and prototyped and proposed through the CFI Semiconductor Wafer Representation (SWR) Working Group. Current research is focused on three-dimensional representations.

An information model for semiconductor manufacturing processes has been proposed through the CFI working group on semiconductor process representation (SPR). A pilot implementation of SPR is being built at TMA and an early version is in use in the Sematech Technology CAD Workbench.

2.11.3 Advanced Process Simulation and Design Environments

We are looking at higher-level architectural issues, such as the interrelationships between the framework data representations and the connection of compliant tools to achieve end-user design objectives. We are also investigating the larger questions of the relationship between frameworks for different related domains (e.g., circuit CAD and TCAD), and the integration of design frameworks into frameworks for computer integrated manufacturing (CIM).

Commercial one- and two-dimensional process simulators have been integrated into the CAFE system through a simulation manager interface to the process flow representation (PFR). Full two-dimensional physical simulation of the MIT CMOS baseline process and research processes have been performed from CAFE.

Through a description formalism for device structural and behavioral goals, we hope to be able to extend traditional process and device simulation further towards actual design. Such goals may be direct structural goals; e.g., junction depth, sidewall slope, or they may be electrical, mechanical, or thermal goals; e.g., threshold voltage or impact ionization current at a specified bias.

2.12 Modeling of Advanced Device Structures

Project Staff

Professor Dimitri A. Antoniadis, Keith M. Jackson, Jarvis B. Jacobs, Michael B. McIlrath, Nadir E. Rahman

The design of complex, scalable microsystems at the leading edge of technology requires a combination of theoretical understanding, numerical modeling, and experimental physical data. Atomistic, first principles models, while of great intrinsic value and an essential analytical tool, are not sufficient for designing large, high-performance systems. Designers need the means to examine the space of available technologies to match the specific requirements at hand and make appropriate engineering

tradeoffs in order to select from a range of technology variants.

In order to help meet this need, two ideas are being investigated: inverse modeling and knowledge-based simulation.

By inverse modeling, we mean the "reverse engineering" of process effects, not directly observable, via studies of experimental electrical measurements in combination with other directly observed or independently known data; e.g., physical process parameters. Through careful choice of physical experiments, inverse modeling can be used to calibrate forward models and provide technology designers with powerful tools for both understanding and design synthesis. Inverse modeling techniques can be used both to find device structures that meet desired performance criteria and to help map out the design space of a given process technology, whether the underlying physical phenomena are fully or only partially understood.

For example, the electrical performance and characteristics of extreme submicron, shallow junction transistors are critically dependent on the exact two- and three-dimensional shapes and locations of doped regions of the semiconductor. In such technologies, there is a trade-off between, for example, current drive and the device short-channel effects of drain-induced barrier lowering and punchthrough. Understanding the exact nature of the implant doping profiles is essential for design in the submicron regime.

However, fabrication of such shallow junction devices using preamorphization acceptor implants, e.g., indium, has made determination by forward simulation of the doping profile near the drain/source junctions more difficult. In the case of indium, the basic implant statistics and diffusion coefficients used in simulators are not as well-characterized as are the data for, e.g., boron and arsenic. Development and calibration of an adequate forward model by direct methods would require a large number of costly, difficult experiments, making aggressive exploitation of the technology prohibitively expensive.

Inverse modeling provides an alternative methodology to determine doping profiles. We have been exploring new methods for the determination of the two-dimensional doping profiles for a sub-0.1 micron MOSFET with super-steep retrograde channel doping and ultrashallow source/drain extension structure. Utilizing one-dimensional experimental doping profile data in combination with measured device electrical data, we have found that it is possible to generate a two-dimensional doping profile which matches the experimental

observations. To show that the resulting doping profile is unique, additional extracted doping profile data at the center of the channel for devices of differing channel lengths are incorporated into the analysis.

Knowledge-based simulation methods are being investigated in order to achieve full or partial automation of inverse modeling procedures. Knowledge-based simulation techniques combine numerical computing, symbolic computing, and formal reasoning methods. Through such techniques, the use of theoretical models, designer intuition, computational simulation experiments, and physical experiment data can be integrated in a systematic fashion.

Our approach is to develop applications that employ existing and new representations in computer-aided technology design; e.g., wafer, process, and device representations, and incorporate theoretical and experimenter knowledge bases, as a power assist to designers in the solution of specific problems. These applications create and evaluate computational experiments, calling on existing commercial or other simulators wherever possible.

Both heuristic and purely numerical methods can be used in conjunction with knowledge-based inferencing in order to drive simulations and suggest physical experiments needed for more data. Such techniques help designers gain insight into system and technology design problems and extend traditional process and device simulation towards actual design synthesis.

2.13 Semiconductor Process Repository

Project Staff

Professor Duane S. Boning, Michael B. McIlrath, William P. Moyne, Chantal E. Wright

The goal of this research task is to create a system to facilitate distributed process research and design. Such a system will allow users to retrieve and examine process flows from multiple process libraries across the network.

An experimental implementation of a networked semiconductor process repository has been demonstrated jointly with Stanford University, using the MIT CAFE database. Via the web, users can select from multiple process libraries, browse process catalogs within libraries, and examine specific processes in detail. Auxiliary software, now being implemented in the java programming language, enables users to assemble their own processes

graphically using repository processes as building blocks. Java is a cross-platform language; programs can be written that will run inside browsers that support java. We make use of the World Wide Web and the java programming language so that the system will be accessible to any potential user on the network. With java, we can produce a system that will not require any specialized software installation by the user. Java's capability of interaction with the network will allow our system to make available multiple, remote process libraries or databases.

Communication and usage of multiple databases requires a common data representation; our system will implement the current standard, the Semiconductor Process Representation. Conversion from the representations used by local systems (such as MIT's CAFE) will be necessary.

In the CAFE system used at the MIT Microsystems Technology Laboratories, process designers (students and staff) construct their own processes by combining process steps from a baseline with experimental processes or customized operations. For the repository, catalog and library (catalog container) objects were designed and implemented in the CAFE database. While the particular organization into libraries (experimental and baseline) and catalogs (thermal, etch, implant, etc.) in this initial repository implementation reflects current usage at MIT, it is not imposed by the design. Web service is provided via a database client program which creates and caches html pages.

Individual processes are presented according to the information model designed through the CAD Framework Initiative and Sematech working groups on process representation. The CAFE process representation is largely a subset of this information model; however, some interface translation is required. In addition to the standard process views effects, environment, equipment, and process (sequence), an encapsulated view accesses the corresponding CAFE process object directly, using a web common gateway interface program. In particular, this view allows examination of the dynamic usage of the process in the CAFE CIM system including current wafer lots, recent measurements, etc.

An optional, experimental feature under development is a web-integrated graphical process editor, adapted almost entirely from one in local use in CAFE. The editor software is written in java. Individual processes from the repository can be loaded into the editor via the web browser. Conversely, clicking on a process instance in the editor will bring the web page for that process from the repository into the browser.

2.14 Process Capabilities Database

Project Staff

Professor Duane S. Boning, Michael B. McIlrath

As semiconductor manufacturing becomes increasingly expensive, few institutions can maintain complete processing capability at the leading edge of research. We have built an information entry and retrieval system for data about university fabrication facilities collected by the Semiconductor Research Corporation (SRC). The system is accessible through the World Wide Web (WWW). Through a web-based query interface, remote users can find university facilities with particular fabrication capabilities and resources. Owners of remote fabs in the database can update information about their own facility.

2.15 National Infrastructure for Networked Design and Prototyping

The advanced computational prototyping tools and methodologies described above both depend upon and can be used to create an infrastructure for virtual and physical prototyping. The quasi-empirical simulation approach requires data from select experiments to build accurate performance, reliability, and variation or statistical models. Parameters for these models must be determined and refined by multiple series of both simulations and repeated experiments. The validation of simulations produced by the computational prototyping tools can only be achieved by correlation of simulated results with actual experimental measurements. Simulations based on the achieved calibrated models can then facilitate subsystem design and physical prototyping.

Some of these experiments may require specialized equipment which is unique or only available at few sites. For example, a suitable chemical mechanical planarization (CMP) machine may not be available at the primary site used by the simulation model developer; but all of the other processing equipment might be easily accessible. Thus, the timely conduct of repeated calibration experiments may be facilitated by local machine processing for all but one step, with that step being accomplished at a remote site.

Additionally, processing and metrology equipment for such experiments may well not be simultaneously available at a single site. Thus, appropriate metrology equipment may also be a scarce resource. The capture, exchange, and use of

experimental data both to develop models and to calibrate or validate these models can be greatly facilitated by utilization of emerging information technologies.

When multiple locations have the same processing or metrology equipment, the availability of such equipment will vary from site to site; and the timely completion of such experiments can be enhanced by arranging for the conduct of the experiment to be split across multiple sites. Furthermore, such experiments will invariably be complicated, requiring many splits.

Coordination of simulation, processing, and metrology resources will enhance collaborative efforts on subsystem design and subsequent physical prototyping. Models calibrated by experimental data collected at multiple sites can then be used at simulations making use of resources at multiple sites to facilitate the collaborative design of subsystems and the following physical prototyping.

It is also necessary to apply the best available computational prototyping capabilities, from MIT and elsewhere, to evaluate novel technology and microsystem architectures. This demands creation and use of the National Research Enterprise (NRE) to make possible distributed process and model repositories, computational tool executions, and collaborative design of experiments and microsystems. We will partner with Stanford University and other research sites to make the NRE a reality by extending our existing design and fabrication support systems to take advantage of National Information Infrastructure (NII) and high performance communications capabilities. The resulting infrastructure will thus support both computational and physical prototyping and greatly enhance the ability to explore new technologies and microsystems.

Our approach will be based on the use of the semiconductor process representation (SPR) as the vehicle with which to specify the change in wafer state goals, the treatment level parameters, the actual processing instructions, and the measurements to be made. The infrastructure that we propose will be designed to facilitate the accommodation of multiple CIM systems in operation across the country. Through the use of the SPR, simulation results will be unified with actual processing results. Thus, it is easier and more efficient to conduct experiments which can be used to validate the simulation results or, more usually, to calibrate the simulator so that the simulation results can be believable and, therefore, considerably more useful in the investigation of new process technology.

Remote fabrication is a technique that can improve the flexibility and efficiency of current integrated circuit fabrication technologies. Prototyping and experimentation can be much more robust, quicker, and more cost effective by providing the capability to perform different wafer processing and metrology steps at different facilities. Expensive and/or unique machines can be shared rather than duplicated. Wafers can be rerouted to different facilities upon protracted equipment failures. Currently, the process of remote fabrication is discontinuous and unorganized; there is no clean or structured way to exchange data or to provide feedback among multiple facilities.

The proposed development of remote simulation, fabrication, and inspection infrastructure will facilitate processing and experimental data interchange, remote inspection techniques, management of reporting mechanisms for assessing the current state of each experiment in progress, as well as collaborative microsystems design and physical prototyping.

2.16 Metal Reliability and Electromigration

Project Staff

Yonald Chery

As the minimum feature size of microelectronic devices continues to decrease, designing against failure inducing phenomena such as electromigration becomes increasingly important. Electromigration is current induced diffusion in metal interconnect. Electrons transfer momentum to metal atoms, causing an atomic flux in the direction of the electron flow. This effect is proportionally related to current density, which tends to rise as integration densities increase.

Recent research has demonstrated that interconnect reliability is strongly dependent not only on the interconnects current density stresses, but also on the distribution of the metal crystal grain sizes in the metal film from which the interconnect is patterned. Due to such microstructural inhomogeneities (e.g., the distribution of interconnect crystal grain sizes), flux divergences in the presence of current densities can occur. At such sites, the depletion of metal interconnect can form a void or accumulate and yield a short to some neighboring interconnect. Both effects can result in the eventual functional failure of some sub-circuit dependent on the failed metal interconnect.

Such physically-based, microstructural interconnect failure models make it possible to more accurately

predict electromigration induced failure. Prior to development of such models, one of the most popular mechanisms, such as Berkeley Technology Associates' BERT tool, required fabrication of test structures representative of one's design. Data collected from experiments on these structures in turn used to calibrate a failure model. However, these newer models have the distinct advantage in that they are a function of process parameters (e.g. metal film grain size distribution), which provides design independence and eliminates the need for test structure fabrication.

Our current research goal involves the development of a computer-aided design tool to provide electromigration reliability feedback to the circuit designer. By using layout and circuit stimulus information along with these microstructure based reliability models, it is possible to compute the reliability distribution of the metal interconnect.

Research is well underway into investigating the requirements for an electromigration reliability CAD tool. Work has already begun on implementing our current tool framework. Our plan for a prototype electromigration tool consists of parsing layout files emitted from MAGIC to extract interconnect geometries. The next two phases involve deducing possible current directions. This information on the currents would be then used to tile the metal interconnection trees with a primitive set of microstructural electromigration geometry abstractions. At present, the abstraction set only provides a model for straight metal interconnect segments, but as further electromigration experiments are performed, other model primitives for interconnect shapes such as "L", "T"s, and "+" will be added. Once tiling the interconnect with these abstractions is performed and the current stresses and directions are known, the appropriate electromigration model can be used and a measure of failure computed.

Development of the first parsing phase of this tool is complete, and we expect to progress rapidly through the other prototype's modules. Concurrently with development efforts on this tool, experiments and further modification and development of an existing metal grain growth simulator is underway for understanding how to characterize different reliability of the different tiling geometry primitives. To develop a larger circuit designer, further work is needed to address the tool's computational efficiently.

2.17 RTFM: A Digital Design Lab Expert

Project Staff

Owen Wessling

This thesis describes the design and construction of the resource teaching framebased machine (RFTM), a knowledge based (or expert) system which helps users to debug and design systems. The RFTM system seeks to act in a more educational manner rather than as a direct problem solver, as it is intended for primary use within introductory digital design courses. The project, while seeking to help users with general problems when asked, is primarily based upon problems and questions students are likely to encounter in the 6.111: Introductory Digital Systems Laboratory class at MIT.

2.18 Publications

2.18.1 Journal Articles

Bonvik, A.M., and Y. Dallery, and S.B. Gershwin. "Approximate Analysis of Production Systems Operated by a CONWIP-finite Buffer Hybrid Control Policy." Submitted for publication.

2.18.2 Internal Publications

Bonvik, A.M., S.B. Gershwin, and D.E. Troxel. "Operating High Variability Manufacturing Systems." CIDM Memo 95-4. MIT, 1995.

Bonvik, A.M. "A Distributed Discrete Event Simulation, or How to Steal More CPU Cycles Than You Could Ever Imagine." CIDM Memo 95-6. MIT, 1995.

Carney, J.C. "Message Passing Tools for Software Integration." CIDM Memo 95-7 (May 1995).

Carney, J.C. "Message Passing Tools for Software Integration." CIDM Memo 95-11. MIT, 1995.

Fischer, G., and T. Lohman. "Error Handling within CAFE." CIDM Memo 95-1. MIT, 1995.

Kao, J. "Remote Microscope for Inspection of Integrated Circuits." CIDM Memo 95-5. MIT, 1995.

Kao, J. "Remote Microscope for Inspection of Integrated Circuits." CIDM Memo 95-10. MIT, 1995.

Kao, J., and D.E. Troxel. "MIT Remote Microscope." CAPAM Memo 95-1. MIT, 1995.

Kwon, J.Y. "Remote Fabrication of Integrated Circuits." CIDM Memo 95-3. MIT, 1995.

Kwon, J. "CAFE Remote Fabrication System—A Quick Step Guide to Remote Processing." CIDM Memo 95-8. MIT, 1995.

Kwon, J. "Remote Fabrication of Integrated Circuits—Software Support for the M.I.T. Computer Aided Fabrication Environment." CIDM Memo 95-9. MIT, 1995.

Moyne, W.P. "Run-by-Run Control: Interfaces, Implementation, and Integration." CIDM Memo 95-2. MIT, 1995.

Rahman, N. "A Survey of Existing Literature on Reverse Modeling of Field Effect Transistors." CAPAM Memo 95-3. MIT, 1995.

2.18.3 Meetings Papers

McIlrath, M. and D. Boning. "Process Repository." 1995 TCAD Symposium. *Hierarchical Technology CAD—Process, Device, and Circuits*. Stanford University, Palo Alto, California, August 10, 1995.

Tso, K., M. McIlrath, and D. Boning. "Fab Facilities Database." 1995 TCAD Symposium. *Hierarchical Technology CAD—Process, Device, and Circuits*. Stanford University, Palo Alto, California, August 10, 1995.

2.18.4 Theses

Carney, J.C. *Message Passing Tools for Software Integration*. S.M. thesis. Dept. of Electr. Eng. and Comput. Sci., MIT, 1995.

Jacobs, J. *Modeling of Electron Transport in Sub-100 nm Channel Length Silicon Mosfets*. Ph.D. thesis. Dept. of Electr. Eng. and Comput. Sci., May 24, 1995.

Kao, J. *Remote Microscope for Inspection of Integrated Circuits*. S.M. thesis. Dept. of Electr. Eng. and Comput. Sci., MIT, 1995.

Kwon, J.K. *Remote Fabrication of Integrated Circuits - Software Support for the M.I.T. Computer-Aided Fabrication Environment*. S.M. thesis. Dept. of Electr. Eng. and Comput. Sci., MIT, 1995.

Moyne, W.P. *Run-by-Run Control: Interfaces, Implementation, and Integration*. S.M. thesis. Dept. of Electr. Eng. and Comput. Sci., MIT, 1995.

Rahman, N. "Mosfet Drain/Source Doping Profile Extraction Using Inverse Modeling." Thesis proposal, Dept. of Electr. Eng. and Comput. Sci., Dec. 22, 1995.

Wessling, O.P. *RTFM: A Digital Design Lab Expert*. M. Eng. thesis. Dept. of Electr. Eng. and Comput. Sci., MIT, December 1995.

