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

Academic and Research Staff

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

Graduate Students

Asbjoern M. Bonvik, John C. Carney, Lynn D. Gabbay, Jarvis B. Jacobs, James T. Kao, Jimmy Y. Kwon, Nicole S. Love, William P. Moyne, Miltiadis A. Stamatopolous, Evren R. Unver

Undergraduate Students

Victor E. Chin, Nadir E. Rahman, Kit Yan Tso

Technical and Support Staff

Francis M. Doughty

2.1 Introduction

Sponsors

Advanced Research Projects Agency
U.S. Navy
Contract N00174-93-K-0035

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.

2.1.1 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 very same process flow representation.

2.2 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.

Current work involves the implementation of hierarchical scheduling, development of factory design tools, real time process control using the PFR, and integration of a TCAD system using the PFR.

2.3 Organization of Data and Algorithms for Factory Design and Control

Project Staff

Miltiadis A. Stamatopolous, Stanley B. Gershwin, Professor Donald E. Troxel

A methodology for the design, redesign, dynamic reconfiguration, and operation of production systems is proposed. A set of tools and data abstractions provide aid in decision making for the whole life cycle of factories.

An important aspect of our approach is the use of common data and, when appropriate, common code, both in the analysis and simulation as well as in the operation of production systems. This way, the analysis and simulation are more realistic; and the tools provided for the operation are extensively tested before they are put in use.

In order to effectively tackle such problems as factory design, redesign, dynamic reconfiguration, and operation of production systems, it is necessary to have an efficient way of organizing all the relevant data. This data can be abstracted in information about *where* to produce, *how* to produce, as well as *what* and *how much* to produce. Furthermore, a performance matrix according to which systems can be evaluated (*how well*) in a consistent manner is required. This information must be specified conveniently and in a way that allows several application programs to access and use the same data. In this way, consistency in different experiments and analytical calculations will be guaranteed.

It is very important to ensure the commonality of the data in all applications that require it. In particular, the same data, as well as scheduling algorithms and code, must be used both in the simulation and in the real time control of production systems. In this way, the simulation will be more realistic; and the scheduling code can be extensively tested before it is put in use.

Several commercial simulation packages exist, but none can guarantee the consistency of data and algorithms used in the design process and in real time control of production systems. In this paper, we develop a methodology that incorporates all stages of factory life, from design to operation. To realize this task, we break it into meaningful abstractions both in data organization and in the procedures used.

2.4 Statistical Process Control Charts for the Computer Aided Fabrication Environment

Project Staff

Nicole S. Love

Statistical process control (SPC) charts are means of improving process performance through the analysis of post-process measurements. SPC charts for the computer aided fabrication environment were developed to assist in the manufacturing of integrated circuits. This thesis describes the application program to produce the SPC charts focusing on four main objectives: generating the control charts, accessing the data from the database, caching the data, and combining control charts. The generation of the control chart is independent of the data and requires a data file of the correct format. The control chart user interface is implemented using the Tcl/Tk scripting language along with [incr Tcl], an extension to Tcl. Accessing the data is accomplished by using the PFR to link a process step with the corresponding inspection step. Caching the data is accomplished using a Gestalt object to store the time frame of the data, the data, and other information used in generating the control chart. The advantages and disadvantages of combining control charts, including multiple machines running the same recipe, and multiple recipes on the same machine, are examined.

2.5 Run-by-Run Control: Interfaces, Implementations, and Integration

Project Staff

William P. Moyne

One of the largest barriers to implementing run-by-run (RbR) control of manufacturing processes is compatibility with existing systems. Manufacturers are reluctant to modify "working" equipment to implement an unproven modification. The RbR controller will attempt to remedy this. The controller will use layered messaging as its interface model. Multiple layers will be designed to interact with external data sources at different levels of abstraction, from objects to primitive scalars. Internally, the controller will use an object based interface to its data as well as a well-defined algorithmic interface. These interfaces are needed for implementation of existing algorithms as well as future algorithmic development. Finally, as a test of the above goals, the RbR controller will be integrated into a variety of systems including both a local CIM system and a control framework being developed as a joint university and corporate effort.

RbR control is a form of process control. It uses post-process measurements and possibly *in situ* data to modify models of the process, and based on these adaptive models recommends new equipment settings for the next run. Although RbR may use *in situ* data, it differs from real time control in that it only adjusts process variables between runs. RbR control can be effective in keeping a process on target and in statistical control in the presence of process noise and drift.

There are many approaches to RbR control. One basic approach is gradual mode control. This method assumes that the actual changes in the process are minor (less than three standard deviations) and only small changes in the control parameters are required in order to compensate. Usually, an exponentially weighted moving average filter is used to distinguish between random noise and real disturbances, thus smoothing the control decisions. This form of control complements existing systems well by suggesting new recipes to the operators of the equipment.

A modified version of gradual mode which was developed at Texas Instruments is predictor corrector control (PCC). This method more explicitly models the time dependence of disturbances. Trends in the drift are monitored over many runs and are factored into the suggested recipes. This method requires the addition of state to the process model, but has shown improved performance over gradual mode when persistent drift is present. An example cause of such drift is equipment aging.

Another approach to RbR control is the rapid mode. This addition integrates control with more traditional statistical process control (SPC) charting and alarm methods. Using statistical rules, the performance of a process is classified as either in-control or out-of-control. Gradual mode control is used for an in-control process, while a more aggressive control strategy (rapid mode) is used to bring out-of-control processes back on target. This provides both rapid correction for statistically large variations such as step changes, while reducing the risk of over control through filtered gradual mode control decisions.

In addition to the methods already discussed, neural networks have also been applied with some success. In this method, a neural network is trained on the inputs and outputs present after a shift has occurred. The controller then uses this information to immediately compensate for disturbances after a shift rather than waiting for convergence in the case of gradual mode. Gradual mode control may then be applied once the shift has been compensated for. Experiments have shown that this method does reduce process variation when compared to pure gradual mode.

The concept of a clean and open interface to the above algorithms has not been studied as well as the algorithms themselves. However, there have been several attempts at defining the form of the data needed for RbR control as well as presenting the controller in a user friendly and easy to use manner. The control algorithms have also been converted to "C".

Incompatibilities between existing equipment and current control implementations are a barrier to adoption of RbR control routines in the VLSI fabrication environment. The RbR controller will attempt to address some of the leading issues blocking existing implementations. In order to communicate with existing systems, the RbR controller will use layered messaging as its interface model. Multiple levels of communication will be possible to provide a wide range of interaction, from complete object interaction to primitive scalar data passing. There will be well defined boundaries between each layer, and translation modules will be written to promote multiple primitive data items into complete objects. The controller itself will use an object based communication model and rely on these translations for interaction with non-object based external sources.

Once the communication model has been defined, the controller will provide a well-defined interface to existing algorithms to ensure that maintenance of code and integration of new algorithms will be as efficient as possible. This interface will embrace ideas such as a programmatic interface, data abstraction, and ANSI "C" coding techniques.

Finally, as a test of the above techniques, the controller will be integrated into an existing CIM system (CAFE), a control framework (GCC), and a locally produced RbR graphical user interface (GUI). These widely varying implementations will test the robustness of the RbR controller's communication model as well as its performance in a wide variety of implementations. It is hoped that by providing a generic and robust model of communication, the RbR controller can become both a test bed for further algorithmic development as well as a model for RbR messaging techniques.

2.6 Hardware Setup for a Remote Microscope

Project Staff

James T. Kao

We have defined the equipment needed to set up a remote microscope (inspection) unit. Although our test unit will not be mounted on a microscope initially, this could be easily accomplished in the future by using a special mount. The hardware for the remote inspection unit basically consists of a PC (connected to the network) installed with a framegrabber board, which will grab images from a video camera.

After capturing images with a framegrabber board, we will save these files to the network using PC-NFS and an ethernet adapter such as the Interlan PCI Ethernet card. The remote microscope user interface software will then be responsible for processing this image and transferring it to remote inspection sites.

Typically in a remote microscope setup, the RGB video signal from the camera's DSUB connector will be sent to the Targa+ DSUB input using a CCXC9DD cable. However, to view the live video image, a separate video monitor must be used. By getting the monitor's video signal from the Targa+ instead of the video camera, one can view the video exactly as seen by the framegrabber board. On the other hand, if one would rather use the camera-monitor setup without going through the computer first, the camera can be attached directly to the monitor. The CCXC9DD monitor has a horizontal resolution (420 lines) that matches the capabilities of the DXC151 camera (440 lines RGB or 460 lines Y/C or VBS); but depending on one's desired applications, a lower resolution (cheaper) monitor might be preferable (for example, if the monitor is used only as a view finder and not as a detailed display). Regardless of the resolution of the video monitor, the frame captured by the Targa+ board will still be very high quality, since the capture resolution of the Targa+ is 640x480.

We are now developing software to control the system. One possible approach is to have the remote inspection site's terminal serve as a client that will connect to a local server that can request grabbed frames from the PC. Another approach would be to have the PC become the actual local server to which a remote client would connect. In either case, the hardware described here (Targa+, camera, lens, power supply, cables, Pentium computer, ethernet card, and video monitor) will still be required in order to provide capability for capturing video frames and passing them through the network.

2.7 First-Order Model for Remote Fabrication

Project Staff

Jimmy Y. Kwon

The first-order processing model makes explicit the information transfer stage during remote processing. The processing of information might proceed as follows. The user writes the process flow text, installs it into the CAFE database, and then creates a new lot and task tree. The local machine operations are then done in the typical operate machine/next operation cycle. When a remote machine operation comes up, the operate machine/next operation cycle breaks in preparation for the remote operation. The first step is the locking of data.

The locking step is needed to provide some measure of data security in the local database. Assume for simplicity that both the local and remote facilities are running CAFE. When a lot leaves the local facility, the database should reflect that state and prevent any changes to the information associated with the lot until the lot returns. This could be thought of as a delayed evaluation. Suppose one wants to compute the product 7*17 on a remote processor. The value 7 is stored in a variable x and the value 17 is stored in y. After storing the values in the variables and sending the computation request, one cannot change the values in x and y until the computation is complete; or else the result one gets will be wrong for the original problem. Similarly in CAFE, appropriate objects in the database should be protected from changes while the lot is away.

We need to consider what sort of objects should be locked in CAFE while operations are done remotely, since in CAFE objects typically contain references to other objects. To begin with, the lot object itself needs to be locked. Lots contain wafer sets, which contain wafers, so these also should be locked. The lot has an associated task tree, so the parts of the process flow and task tree that are sent to the remote facility should also be locked and protected from tampering. In this case, we can lock only the portion of the task and flow tree that involves the remote operations, leaving the other parts free for modification (i.e., the present group of remote operations are locked, but future operations may be modified). Leaf tasks have opinst objects that may also need to be locked.

2.8 Estimating the Lead Time Distribution of Priority Lots in a Semiconductor Factory

Project Staff

Asbjoern M. Bonvik

We consider the lead time of priority lots in a semiconductor factory. The main cause of delays is waiting for repairs of failed machines, and the failures are mainly caused by power disturbances. This can be modeled as an exogenous time-dependent failure process, and we develop a probabilistic model of the system based on this. Using this model, a convolution algorithm for finding the lead time distribution is described.

We describe a method of creating synthetic samples of the lead time from historical failure and repair data collected in the factory. Based on such a sample from the MIT Integrated Circuits Laboratory, we approximate the distribution of lead times by a gamma probability density function shifted by the smallest possible lead time for the product type. The parameters of the gamma distribution are found by using a maximum likelihood estimator. The resulting distribution gives good agreement with the synthetic data for values less than two standard deviations above the mean lead time.

Since our procedure depends only on a description of the process and the failure and repair history of the factory, it can also be used to obtain lower bounds on the lead time for new product types.

2.9 Factory Display Software

Project Staff

John C. Carney

A semiconductor manufacturing facility contains several machines, each of which operates on lots. At any given time the state of a machine may change. The machine might go down for repairs or might receive a new lot on which to operate. Status information of this sort is of great importance to managers and workers of the facility. Factory display software has been developed which makes this information available graphically. By providing the user a map of the facility, with iconic representations of machines, the user can quickly and easily view the status of the facility. In addition, the user can probe the map to obtain other desired information about the machines and lots within the facility.

It is intended that the factory display software will be used in real time, showing the current status of machines and lots with an active display that dynamically changes as the status information changes. However, there are other important uses. The factory display can also be used to "play back" the events of a facility much faster than real time. This would allow a user to perhaps view the entire previous week's events in just a few minutes. The graphical manner in which a user can see all machines at every instant in time can convey an enormous amount of information in a very suitable way. While a view of the facility's historical information is useful, also very useful is a view of simulated information. The factory display software has been designed with this in mind. It can be integrated with facility scheduling and simulation software to provide an "output device" for the data produced by such programs.

The factory display software is written in Tcl/Tk, a scripting language and toolkit for creating graphical user interfaces under X Windows. The display program uses a message passing system to dynamically receive messages containing status information about the facility. The message passing nature of the display program allows it to be easily integrated with other existing or developing systems. The factory display program has successfully been integrated with CAFE.

2.10 Message Passing Tools for Software Integration

Project Staff

John C. Carney

Unix-based environments typically provide a rich abundance of tools. Generally these tools are integrated by operating on a common set of files. There are no interactions among tools and each provides its own interface. One approach for integrating tools is through the use of a message passing system. The goal of a message passing system is to provide a method by which structured information can be exchanged between two or more running processes. These processes may all run on a single workstation or on separate workstations connected by a network.

A flexible peer-to-peer approach to message passing has been developed. This approach allows an enormous variety of connection architectures, where each program establishes one or many connections with various other programs. The communications handling application tools (CHAT) suite has been developed, providing the programmer with a suite of libraries which can be used to integrate software based on the peer-to-peer message passing approach. The suite contains three libraries of routines which provide support for packaging of data into messages, handling of incoming messages, and establishment of connections between programs. All three libraries are written in C and in Tcl/Tk. The CHAT suite will allow for 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. Libraries have also been used to integrate a graphical user interface and simulation environment with an RbR control server. Plans have been made to use the CHAT suite in research work being done on the remote fabrication and remote inspection projects within MIT's computer integrated design and manufacturing (CIDM) project.

2.11 CAFE at MIT Lincoln Laboratory

Project Staff

Gregory T. Fischer

MIT Lincoln Laboratory is using CAFE as the CIM system for use in their IC processing facility. Multiple lots are being processed daily via PFR based fabrication. Lincoln Laboratory is committed to having 100 percent of their wafers in this facility processed with PFR based fabrication.

Release 5.3 of CAFE was installed at both MIT and Lincoln Laboratory. A number of enhancements and improvements have been made to CAFE in response to user requests and feedback both at Lincoln and at our campus fabrication facilities. They are described below. Perhaps most noteworthy is the substantial speed-up of Generate Traveller and Operate Machine.

Generate Traveller has been modified extensively due to previous speed problems and corrupted traveller cache files at Lincoln Laboratory. The traveller report for a PFR lot is precalculated and stored in files for faster generation of a traveller report upon request by a user. In previous releases of CAFE, the initial traveller cache for a PFR lot was calculated the first time a user selected the "Generate Traveller" menu option. For CAFE release 5.3, the initial traveller cache is now calculated at "Start Lot" time, cutting the initial cache creation time by a factor of at least five.

The caching scheme for a traveller report was also altered. The old caching scheme involved up to three files for a PFR lot: a past report file, a present report file, and a future report file. "Generate Traveller" would check the lot to see which operations had been performed since the last traveller report generation and would update the three files accordingly. With CAFE release 5.3, a traveller cache file is created for each top-level opset task in a lot's task tree. These files are stored in a /usr/cafe/cache/traveller/<lotname> directory. Each filename is a number, such as 003; and listing the directory will display the files in the same order as they appear in the task tree. A schema change was made to add an "advice" slot to TASK database objects, providing a place to store the traveller cache filename which belongs to a given top-level opset task.

The updating process for an existing traveller report has also been enhanced. Any function which alters a task's status, such as "Operate Machine", now updates the correct traveller cache file for that task upon exiting. Any function which alters a lot's task tree must delete the appropriate traveller cache file for recalculation at a later date. Upon invoking "Generate Traveller", if only task status has been modified, generation of the traveller report now takes three to five seconds. If a lot's task tree has been altered, only the top-level opset tasks which have no traveller cache file are updated before the full traveller report is generated. This takes about half the time it took to update the traveller cache in previous versions of CAFE.

Enhancements to the speed of "Operate Machine" have also been performed. The functions which calculate the WAFERSET and/or WAFER display have been optimized. Also, at "Start Lot" time, the Fabform user interface screen for each fringe task is calculated and stored in the PLANNEDOPINST object for that task. In tests on cafe.mit.edu and garcon.mit.edu, these enhancements have dropped the time to get into "Operate Machine" from twenty seconds to around five seconds. The time to get out of "Operate Machine" has been slightly retarded due to traveller cache file updating, seven seconds in the previous version of CAFE to twelve seconds in CAFE release 5.3.

"Operate Machine" and "Next Operation" have been modified to allow users to continue from one "Operate Machine" to the next without getting back to the "Fabrication Tools" menu. Once an operation is completed in "Operate Machine", a message is displayed informing the user of the next machine operation to be performed; and the user is asked if he wishes to indeed operate the next machine. If the response is "yes", "Next Operation" is called. If there is only one machine choice for the next operation, "Operate Machine" is immediately entered again for that machine. If more than one machine is valid for the next operation, a cleaner "Next Operation" screen is presented which allows the user to choose which machine to operate.

"Start Lot" has been modified to check a lot's wafer sets against the wafer sets specified by the PFR process. If a lot is found to be lacking wafer sets needed by the process, an error message is displayed informing the user as to which wafer sets are missing. "Start Lot" also generates the initial traveller cache for the lot.

New trend chart applications are now under the "Reports" "Trend" menu. Tcl/Tk graphical user interfaces allow users to design, edit, display and plot SPC charts for PFR-based diffusion processes which have been performed.

Traveller caches for all PFR lots are now updated overnight via backgrounded cache update functions. Invoking these functions during nightly cafe maintenance will insure up-to-date traveller reports each morning.

A new Tcl/Tk graphical user interface has been developed to replace the current Fabform user interface. The new interface combines graphical menus with Fabform-based application interfaces. Users can toggle between the new interface and Fabform via the "General Purpose" "Change Interface" application.

A new PFR-based process rule checking capability is now available via the "Task Editor". This allows a user to test a process and check for possible problems and safety hazards, such as placing wafers which are still coated with photoresist into a diffusion furnace. Eighteen rules are currently checked.

A new user interface has been developed for the "Modify Machine Status" application which allows users to modify the status of multiple machines within a facility without having to return to the "Labstaff Tools" menu.

A new subclass of the OPINST database object was created: REALOPINST. Subtypes of a REALOPINST are ACTIVEOPINST and COMPLETEDOPINST. The TASK schema was modified to use the REALOPINST in the "opinst" slot. This change will prevent PLANNEDOPINSTs from incorrectly pointing to a TASK's "opinst" slot.

It is now possible to view CAFE Help and Lab Manual text via the World Wide Web and xmosaic. Options have been added under the "Help" and "Help" "Equipment Help" menus which will initiate an xmosaic window displaying the specified help text section.

2.12 Error Handling Within CAFE

Project Staff

Gregory T. Fischer, Thomas J. Lohman

CAFE users were frequently encountering errors within the CAFE software package, both at the MIT Microsystems Technology Laboratories (MTL) and at Lincoln Laboratory. These errors would often result in the user being dumped in to the Lisp debugger and/or having the CAFE session abort completely. After investigation, new methods of catching these errors were developed and incorporated into the CAFE software.

The interception of the Lisp error handlers and the ability to disable CTRL-C signals have been instrumental in tracking and preventing CAFE errors. The frequency of error email messages has dropped dramatically from an average of five a day to only one or two per week, suggesting that the previous user CTRL-C interrupts were causing more wide-ranging problems than was first thought.

2.13 Semiconductor Manufacturing Process Flow Representation

Project Staff

Michael B. McIlrath, Professor Donald E. Troxel, Professor Duane S. Boning, Evren R. Unver

2.13.1 Introduction

The PFR and its integration into both design and fabrication operations is central to the CIDM program and the CAFE software system.[1] CAFE is currently being used at the semiconductor fabrication processing facilities of the MTL, Lincoln Laboratory, 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

[1] M.B. McIlrath, D.E. Troxel, D.S. Boning, M.L. Heytens, and P. Penfield, Jr., "CAFE - The MIT Computer-Aided Fabrication Environment," *IEEE Trans. Comp., Hybrids, and Manuf. Tech.* 15(2): 353-360 (1992).

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, 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 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 software frameworks for TCAD and CIM, including MIT CAFE. Initially a two-stage generic process step model was used, which described processing steps in terms of two independent components: (1) an equipment-dependent, wafer-independent stage, which maps equipment settings to physical processing environments; and (2) 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.

2.13.2 Process Development and Execution

The PFR has been used to develop and execute all of the fabrication processing at Lincoln Laboratory during the last two years. Over five hundred 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 of 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 CIDM and MTL research activities including hot carrier reliability and extreme submicron technology design.

2.13.3 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 here 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 development, to warn a user that an illegal or incorrect process flow has been specified.

Each design rule is a separate entity and can be developed and applied independently. Customized groups of rules can be created. Each rule is a constraint associating wafer state information and process step information. Rules may be based on any process attribute or set of attributes, e.g., wafer state, physical treatment, equipment, etc. The constraint is active just when the process step information matches the attributes of the process step specified in the PFR. If a constraint is not satisfied, an error is signaled. Unsatisfied constraints thus "block" further execution of the process. If the step can proceed (that is, all rules are satisfied) the running wafer state is updated with the necessary information, determined from the process attributes. Rules are applied in parallel in a single walk through the process; it is not necessary to sequentially run multiple design rule checking programs. Splits and joins are supported.

In on-line (fabrication) rule checking, running wafer state information is maintained in the task tree. The on-line wafer state can also be used to verify the correctness of changes made to a flow during processing by inserting, deleting or modifying the nodes of the task tree associated with the flow.

2.13.4 Conceptual Graphs Applied to Semiconductor Processing

Conceptual graphs are known to be relevant to a broad spectrum of problems in knowledge representation, including modeling of physical processes. We have applied the conceptual graph formalism to the general process modeling framework underlying the PFR[2] and are working on the application of conceptual graphs to semiconductor problems including process control and design.

2.13.5 Investigation of PFR Requirements for Factory Design and Real Time Scheduling

We have begun investigation of PFR requirements for factory modeling and design, simulation, and real time scheduling. The goal is to be able to support the development of methods for the design of an efficient factory and the efficient operation of a factory to meet production requirements. Among the needs that have been identified are a more sophisticated and precise definition of operation time, including setup time, initial operator time, unattended operation time, and final operator time; and the proper handling of PFR conditionals for scheduling, including predictive branching.

2.13.6 Extension Language Interface

An extension language provides a programmable way for users to extend an application software tool or framework. An extension language interface to the CAFE system has been implemented and is designed to be compatible with the CAD framework initiative (CFI) extension language (EL), which is based on Scheme. Because the PFR has a programmatic, textual form, designed for extensibility, extension language forms may be embedded in the PFR to access the CAFE database and services and express computations. An extension language intepreter has been integrated into the PFR interpreter, and we are experimenting with the use of this capability in process design and fabrication.

2.13.7 Ongoing Standards Activities

Work is continuing in conjunction with the CFI working group on semiconductor process representation (SPR). A pilot implementation of SPR is being built at Technology Modeling Associates (TMA) and an early version is in use in the Sematech Lithography Modeling Workbench. We are working closely with TMA and Sematech to insure compatibility with larger integrated design and manufacturing framework principles and goals. A formal base information model for SPR, developed largely from MIT work, is currently being considered by the SPR working group and Sematech reviewers.

2.14 Computer-Aided Technology Design

Project Staff

Professor Dimitri A. Antoniadis, Professor Duane S. Boning, Lynn D. Gabbay, Jarvis B. Jacobs, Michael B. McIlrath, Nadir E. Rahman, Kit Yan Tso

[2] D.S. Boning, M.B. McIlrath, P. Penfield, Jr., and E. Sachs, "A General Semiconductor Process Modeling Framework," *IEEE Trans. Semicond. Manufact.* 5(4): 266-280 (1992).

2.14.1 Introduction

The design of advanced integrated circuit microsystems is increasingly 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 activities proceed in parallel: design of (1) the device structures and (2) the process to fabricate them. In general, the 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.

2.14.2 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 generally divided into (1) tools, such as a simulator, which perform some part of an application task, and (2) 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:

- data representations for the objects of discourse in the application domain and their semantics, and programmatic interfaces to those representations; and
- 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.

2.14.3 Process Capabilities Database

As semiconductor manfacturing 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 fabrications in the database can update information about their own facility.

2.14.4 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).

2.14.5 Conceptual Graphs and Manufacturing Processes

Conceptual graphs are known to be relevant to a broad spectrum of problems in knowledge representation, including modeling of physical processes. We have been working on the development and application of conceptual models of semiconductor manufacturing processes, applying the conceptual graph formalism to the representation of state (wafer, physical variables, and equipment, including actuators and sensors) and relationship information. Specifically, we are investigating: (1) descriptions of the causal chain of effects in manufacturing; and (2) a predictor concept for expressing in detail how effects propagate.

Applications of the conceptual graph formalism to semiconductor manufacturing include causal reasoning about manufacturing processes, process synthesis, diagnosis, and control. In particular, the determination of potential opportunities for closed loop process control is becoming more important given the increasing quality and repeatability demands of semiconductor manufacturing. Analysis of the possibilities for control (e.g., of equipment state, physical variables, or intermediate or final wafer state, via open or closed loop) may be made from the paths in the conceptual graph of process effects. Specific controller synthesis requires predictors detailing the relationship between states in the graph.

2.14.6 Determination of Doping Profiles from Electrical Device Data

We are exploring a novel method for the determination of a two-dimensional (2D) doping profile of a MOSFET using inverse modeling. With the advent of shallow junction technologies using preamorphization acceptor implants, e.g., indium, determination via simulation of the 2D doping profile near the drain/source junctions has been made 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 boron and arsenic. Inverse modeling provides an alternative methodology to understand the 2D doping profile. These methods utilize one-dimensional (1D) experimental doping profile data and measured device data. The 1D experimentally extracted long-channel doping profiles and measured device data consist of: 1D channel profile and 1D extension drain profile versus depth into the silicon; and the threshold voltage versus channel length, drain-induced barrier lowering versus channel length, and threshold voltage versus substrate bias as a function of channel length, respectively. We have found that it is possible to generate a 2D doping profile which matches the above data. To show that our result is unique, the experimentally extracted doping profile data at the center of the channel as a function of channel length is being incorporated into this analysis to reduce the chances of error. Different approaches are being explored for automation of the inverse doping profile modeling process, including both heuristic and purely numerical methods. Understanding the exact nature of the implant profiles enhances our ability to design next generation technologies.

2.15 Publications

2.15.1 Journal Article

Giles, M.D., D.S. Boning, G.R. Chin, W.C. Dietrich, M.S. Karasick, M.E. Law, P.K. Mozumder, L.R. Nackman, V.T. Rajan, D.M.H. Walker, R.H. Wang, and A.S. Wong. "Semiconductor Wafer Representation for TCAD." *IEEE Trans. Comput. Aided Design.* 13(1): 82 (1994).

2.15.2 Internal Publications

Antoniadis, D., D.E. Troxel, D. Boning, S. Gershwin, M. McIlrath, and J. White. "ARPA HPC/Microsystems PI Meeting - Poster Session and Demonstrations." CIDM Memo 94-14. MIT, 1994.

Bonvik, A.M. "A Program for Historical Sample Path Analysis of Cycle Times in Semiconductor Manufacturing." CIDM Memo 94-3. MIT, 1994.

Bonvik, A.M. "Design Proposal for the CAFE Scheduling Application." CIDM Memo 94-6. MIT, 1994.

Bonvik, A.M. "Estimating the Lead Time Distribution of Priority Lots in a Semiconductor Factory." CIDM Memo 94-13. MIT, 1994.

Bonvik, A.M. "Introduction to the CAFE Scheduling and Simulation Testbed." CIDM Memo 94-11. MIT, 1994.

Carney, J. "An Implementation of a Layered Message Passing System." CIDM Memo 94-15. MIT, 1994.

Fischer G. "New Traveller Generator and Operate Machine Speed Enhancements." CIDM Memo 94-8. MIT, 1994.

Fischer, G. "Release 5.3 of CAFE (Version 3.0)." CIDM Memo 94-10. MIT, 1994.

Gabbay, L. "Assessment of Caesar v1.1.0." CIDM Memo 94-9. MIT, 1994.

Gabbay, L. "Backgrounded Lot Traveller Report Caching." CIDM Memo 94-2. MIT, 1994.

Kwon, J. "First-Order Model for Remote Fabrication." CIDM Memo 94-18. MIT, 1994.

Kao, J. "Hardware Setup for Remote Microscope." CIDM Memo 94-19. MIT, 1994.

Kwon, J. "Zeroth-Order Model for Remote Fabrication." CIDM Memo 94-17. MIT, 1994.

Kwon, J., and J. Kao. "Remote Fabrication/Remote Inspection - Work in Progress." CIDM Memo 94-16. MIT, 1994.

Love, N. "SPC Charts for CAFE." CIDM Memo 94-4. MIT, 1994.

Stamatopoulos, M., S.B. Gershwin, and D.E. Troxel. "Organization of Data and Algorithms for Factory Design and Control." CIDM Memo 94-12. MIT, 1994.

Stamatopoulos, M., D.E. Troxel, and S.B Gershwin. "Microsystems Factory Representation." CIDM Memo 94-7. MIT, 1994.

Troxel, D.E. "Caching of Data Base Objects in CAFE." CIDM Memo 94-5. MIT, 1994.

Troxel, D.E. "Tasks and PFR based Fabrication." CIDM Memo 94-1. MIT, 1994.

2.15.3 Meetings and Presentations

McIlrath, M.B. SWIM Meeting at SEMATECH, Austin, Texas, February 17, 1994.

Troxel, D.E., and M.B. McIlrath. CIM Application Framework Review at SEMATECH, Austin, Texas, May 4, 1994.

2.15.4 Theses

Love, N.S. *Statistical Process Control Charts for the Computer-Aided Fabrication Environment.* S.M. thesis. Dept. of Electr. Eng. and Comput. Sci., MIT, 1994.

Stamatopoulos, M. *A Factory Representation as a Design Tool in a Computer Integrated Manufacturing Environment.* S.M. thesis. Dept. of Electr. Eng. and Comput. Sci., MIT, 1994.

Unver, E.R. *Implementation of a Design Rule Checker for Silicon Wafer Fabrication.* M.E. thesis. Dept. of Electr. Eng. and Comput. Sci., MIT, 1994.

2.15.5 Conference Proceedings

Boning, D.S., and M.B. McIlrath. "Conceptual Graphs and Manufacturing Processes." Second International Conference on Conceptual Structures, College Park, Maryland, August 15, 1994.