



12

DTIC FILE COPY

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

VLSI PUBLICATIONS

COMPUTER-AIDED FABRICATION SYSTEM IMPLEMENTATION

Semiannual Technical Report for the period October 1, 1986 to March 31, 1987

Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

Principal Investigators:	Paul Penfield, Jr.	(617) 253-2506
	Dimitri A. Antoniadis	(617) 253-4693
	Emanuel M. Sachs	(617) 253-5381
	Donald E. Troxel	(617) 253-2570
	Stanley B. Gershwin	(617) 253-2149

This research was sponsored by Defense Advanced Research Projects Agency (DoD), through the Office of Naval Research under ARPA Order No. 5339, Contract No. N00014-85-K-0213.

AD-A179 450

DTIC ELECTED
S APR 22 1987 **D**
D

**APPROVED FOR PUBLIC RELEASE
DISTRIBUTION UNLIMITED**

87 4 21 012

Microsystems
Research Center
Room 39-321

Massachusetts
Institute
of Technology

Cambridge
Massachusetts
02139

Telephone
(617) 253-8138

TABLE OF CONTENTS

Research Overview 1
 CAF System Structure 2
 Modular Process 4
 Equipment Modeling 6
 Scheduling 8
 Publications List 9

Selected Publications (starting after page 10)

T.-L. Tung, J. Connor, and D. A. Antoniadis, "A Viscoelastic BEM For Modeling Oxidation," Proceedings of NUMOS I, An International Workshop on Numerical Modeling of Semiconductors, Los Angeles, CA, December 1986. Also, MIT VLSI Memo no. 87-371, March 1987.


Sheldon X. C. Lou, Garrett Van Ryzin, and Stanley B. Gershwin, "Scheduling Job Shops with Delays," to appear in Proceedings, 1987 IEEE International Conference on Robotics and Automation, Raleigh, NC, March 31 - April 2, 1987. Also, MIT VLSI Memo no. 87-369, March 1987.

Oded Z. Maimon and Stanley B. Gershwin, "Dynamic Scheduling and Routing for Flexible Manufacturing Systems that have Unreliable Machines," to appear in Proceedings, 1987 IEEE International Conference on Robotics and Automation, Raleigh, NC, March 31 - April 2, 1987. Also, MIT VLSI Memo no. 87-370, March 1987.

P. Penfield, Jr., "Computer-Aided Fabrication of Integrated Circuits," invited talk, Advanced Research in VLSI, Stanford University, Palo Alto, CA, March 23-25, 1987. (slides included from presentation)

Accession For	
NTIS CRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail and/or Special
A-1	






RESEARCH OVERVIEW

The work under this contract is carried out in four areas. These are concerned with CAF system structure, process simulation and modeling, equipment modeling and simulation, and scheduling.

Our CAF system, named CAFE (Computer-Aided Fabrication Environment) is in constant use in the MIT IC Laboratory. It currently runs on a VAX 785, but will shortly be ported to a Sun Microsystems computer. At this time the data base will be converted so it uses INGRES running on the Sun.

A data model for VLSI fabrication facilities is currently under development. There are various type extensions, and the schema captures several important aspects of plant and process management.

Progress has been made on the definition of a language for specifying process flows. This work, done in loose cooperation with the Berkeley CIM effort, is not yet at a stage where it can be used. However, to facilitate operation in the IC Laboratory a very abbreviated and preliminary version of a flow language has been implemented, in the sense that several interpreters have been written that act on the CMOS baseline process under development. The language effort is based on the two-stage generic process-step model which has been discussed in previous reports. This model also forms the foundation for the flow language effort at Berkeley and Stanford, as reported at the informal Workshop on Process Specification held at the conference on Advanced Research in VLSI, March 24, 1987, at Stanford University.



Two-dimensional localized thermal oxidation has been modeled using a visco-elastic boundary element method. This model predicts higher than expected stress during oxidation, and hence possibly plastic deformation.

There is a need for an interchange format for wafer state. We have developed a suite of routines to work with the proposed standard profile interchange format.

The equipment modeling task is a new one. The objective is to develop executable machine models to be used for machine simulation. The models will deal with both nominal values and variances. A matrix experiment will be devised to test such models for LPCVD.

A hierarchical framework for scheduling has been devised, in which different levels deal with different time horizons. The lowest level is the actual, detailed set of schedules. An equipment reservation system for this lowest level has been written and incorporated into CAFE, and is now in daily use.

CAF SYSTEM STRUCTURE

Last year an initial, rudimentary CAF system was designed and implemented. This included a personal notebook and data structures for the upcoming fabrication equipment installation.

Our present computer hardware consists of a VAX 785 with 16 Mbytes of main memory, five 450-Mbyte disks, GCR tape, 1600 cpi tape, two laser printers, four phone lines, and an Ethernet port to the MIT network. Additionally there are eight 11/73-based terminal concentrators, six with 48 RS232 ports, one with 16 ports and one spare for maintenance. RS232 cables have been installed throughout our building, and numerous user terminals and PCs have been connected.

Because of our success in getting faculty, staff, and students to use our computing facilities, and because of the increased use of our CAF system in actually operating the IC Laboratory, our VAX 785 (CAF.MIT.EDU) is heavily overloaded. We must augment our facilities. We evaluated several possibilities and have initiated the purchase of two Sun Microsystems 3/280 computers. One of these will be used for the actual running of the CAF system CAFE for the IC Laboratory and the other will be reserved for development of CAF system software. We also have ordered RTI INGRES for use on the Sun computers. Delivery of this equipment is expected during the first half of 1987.

We have made substantial progress in the development of a data model and schema. Our data architecture being developed is similar to the Multibase system developed at CCA. This provides a uniform query interface to data residing in multiple autonomous, heterogeneous data bases. Our current data base is distributed across two relational systems, university INGRES and PRELUDE and a hierarchical file system. PRELUDE is a fast, lightweight UNIX-style data manager. Our system is very modular and, thus far, it has been easy to incorporate new DBMSs into the system as well as move data from one data manager to another.

Our data model is the functional model with support of extended data types including various temporal types as well as inexact, interval, and null values. The schema captures several important aspects of plant and process management: fabrication facilities and equipment, users, equipment reservations, lots, lot tracking, wafers, process flow descriptions, wip tracking, and lab activity information.

We have developed a generalized forms based user interface program, called fabform. This single program, when called with a parameter file, produces a terminal display and allows a user to move from field to field and enter data. The type and content of the screen display is specified by an ASCII file which is referenced by data included in the parameter file. The form may have arbitrary length and the user can scroll up or down. At present, user interface commands are much like EMACS commands. When the user exits, or saves the data, an updated parameter file is written.

Using fabform, we have implemented the electronic equivalent of a sign-up sheet for reserving fabrication equipment in our lab. This program is in daily use and stores the sign-up data in our data base. Several improvements for the next version of this reservation program have been initiated.

We are developing a process flow language. This language has a lisp-like form, although this need not be apparent to process design engineers. We continue to embrace the concept of the two-stage generic process model with corresponding expression of goals, wafer environment, and machine settings. Adequate machine models, yet to be developed, will provide a transformation from machine settings to wafer environments. Process models similarly provide the transformation from the wafer environment to the wafer-state goals.

The meaning of our process flow language is to be provided by several interpreters: fabrication, simulation, production scheduling, and "walk-through." We have completed an initial version of this "walk-through" interpreter. It interprets the process flow language and enables a process developer or potential user to see what will happen when the process is executed. It will, we think, prove useful in the design of a process, the communication of the process to others, and its approval by laboratory management. Substantial work has been accomplished towards a fabrication interpreter.

We have defined and begun work on a browser which allows a user to review what actually happened in the fabrication of a lot of wafers. We have yet to resolve how to browse through the future, especially when the future path is uncertain due to possible branches.

MODULAR PROCESS

The implementation of the first version of MASTIF (MIT Analysis and Synthesis Tool for IC Fabrication) was completed in May 1986. A menu- and window-oriented program has been developed as the first step in meeting the need for an integrated process design system. MASTIF includes aids for process specification and simple process verification, and provides interfaces (both front and back ends) to the SUPREM-III process simulator and the MINIMOS device simulator. Drawing on analogies from other areas of VLSI design, a blueprint for future development of tools beyond simple simulation has been completed [1]. MASTIF was successfully ported to run on a true workstation, the Vaxstation-II under VMS. The program is now in its second phase of development and in the process of being ported to a UNIX environment.

We have continued our effort on modeling two-dimensional (localized) thermal oxidation. Thermal oxidation of silicon involves the diffusion of oxidant species from the gas-oxide interface to the oxide-silicon interface, and the transport of newly formed oxide away from the latter. Under suitable formulations, it can be shown that the diffusion process is a Laplace problem and the viscoelastic flow of oxide is a biharmonic problem. For these boundary value problems, the unknown boundary parameters can be obtained from the known boundary conditions without calculating the interior solutions. The diffusion problem is solved with a standard boundary element method (BEM) for potential problems. A generalized viscoelastic BEM has been developed to model the oxide flow. Utilizing constant-velocity kernel functions, this viscoelastic BEM can deal with a wide range of stress relaxation times, covering elastostatic deformation and incompressible creeping flow. Our approach achieves simplicity and efficiency by solving a two-dimensional problem as line integrals on the boundaries. Simulations of Local Oxidation of Silicon (LOCOS) structures indicated that stress created during oxidation could be extremely high, particularly when the mechanical barrier effect of silicon nitride mask is included. This suggests that both silicon nitride and oxide flow or plastically deform more readily than assumed. Stress-induced retardation of reaction rate and diffusivity of oxidants are also studied. In addition to an overall lowering of stress, the shape of the oxide changes significantly when such nonlinear effects were included.

One important finding of the MASTIF project has been to note the need for a uniform representation of wafer and device structure information, both for use by individual tools in a complete design system, and for interchange between different simulation sites. Development of general analysis tools and interfaces requires a central, agreed-upon representation of the structures these simulators and associated tools manipulate. We have been heavily involved in standards work related to a Profile Interchange Format (PIF), and are implementing a library of routines for accessing these structures based on the PIF [2]. Specific tools and components under development include SNC, a local "database" form for PIF storage, PIFLIB, a library of routines for tool inter-

face to the database, and PIFPLOT, a general, interactive structure analysis tool interacting with the PIF database. It is expected that the availability of a general representation for process and device structures will greatly enhance the capabilities of the MASTIF workstation, and will spark development of additional tools to aid in the design of IC processes.

References:

- [1] D. S. Boning, "MASTIF - A Workstation Approach to Integrated Circuit Process and Device Design", M.S. Thesis, MIT, May 1986.
- [2] D. S. Boning and T.-L. Tung, "A Proposal For A Profile Interchange Format", CAF Working Papers, MIT, April 1986.

EQUIPMENT MODELING

This project is a new one, not included in previous reports.

VLSI machine modeling has taken a concrete form during the past several months. The thrust of the work is to combine analytical modeling with matrix experimental approaches and to provide an executable program which will facilitate the following:

1. Off line quality control to determine the point in operating space which provides the greatest robustness against variations in process parameters and therefore yields the most consistent results.
2. On line quality control used to tune a process based either on measurements made after a previous run or on in-situ measurements.

Matrix experimentation is a collection of methods wherein some or all of the relevant process variables are varied simultaneously in an experiment and information is extracted from the results statistically [1,2]. In contrast to conventional single-variable experimentation, the matrix approach offers a tremendous economy of experimental effort. This is crucial in a production environment as the interruption to work flow must be kept to a minimum, while in a research environment it is useful in order to optimize a new process as quickly as possible. Matrix experimental techniques have been employed with great success on VLSI processes at AT&T Bell Laboratories for approximately the last six years [3,4].

Our process modeling effort will begin by utilizing analytical and experience based background to define the process variables and their values for a matrix experimental approach. The analytical work will be based both on process models available in the literature and on simple physical models. Future extensions of the work will attempt to more closely couple the analytical and experimental approaches by using matrix experimentation to verify analytical models, specify numerical values for the analytical models, and also to improve analytical models.

The first process selected for the machine modeling effort is LPCVD of poly silicon. Optimization of thickness uniformity across a wafer, between wafers in a single lot, and between runs will be the goal.

References:

- [1] G. Box, W. Hunter, and J. Hunter, "Statistics for Experimenters and Introduction to Design Data Analysis and Model Building," Wiley, 1978.
- [2] G. Taguchi, "Introduction to Quality Engineering," Krauss International Publications, White Plains, NY, 1986.

[3] M. Phadke, N. Kuckar, D. Speeney, and M. Grieco, "Off-Line Quality Control in Integrated Circuit Fabrication Using Experimental Design," The Bell System Technical Journal, 1983.

[4] R. Nackar and A. Shoemaker, "Robust Design: A Cost-Effective Method for Improving Manufacturing Processes," AT&T Technical Journal, March/April, 1986.

SCHEDULING

Research during this period focused on three activities: (1) studying the integrated circuit fabrication process at a systems level, (2) formulating a mathematical model of an integrated circuit fabrication facility, and (3) developing simulation and scheduling software.

The effort to define the scheduling problem continues. We are concentrating on using the MIT laboratories as case studies. Mathematical and simulation models, described below, are being based on what we learn here.

As a mathematical model of an IC fabrication facility, the multiple time scale decomposition under development shows great promise. A new basic model is being investigated which will help us refine and better justify the tentative mathematical results we have developed thus far on hierarchical scheduling.

In this approach, the scheduling algorithm is divided into a set of levels which correspond to classes of events that are distinguished by their frequencies. At each level, two kinds of calculations are performed: small linear programs, to determine frequencies of higher frequency (lower level) events; and simple combinatorial optimizations, to determine exact times for the events of that level, whose frequencies have been calculated at higher levels.

Software which will implement a multiple time scale decomposition approach to hierarchical scheduling is under development. A simulation is also under development. The scheduling software will first be tested with the simulation, and then used to run the laboratory.

An electronic machine reservation system is also under development. In its initial version, it is essentially an electronic sign-up sheet for equipment. It will later be used, after modifications, as the lowest level of the hierarchical scheduler.

PUBLICATIONS LIST

T.-L. Tung, J. Connor, and D. A. Antoniadis, "A Viscoelastic BEM For Modeling Oxidation," Proceedings of NUMOS I, An International Workshop on Numerical Modeling of Semiconductors, Los Angeles, CA, December 1986. Also, to appear as MIT VLSI Memo no. 87-371, March 1987.

Sheldon X. C. Lou, Garrett Van Ryzin, and Stanley B. Gershwin, "Scheduling Job Shops with Delays," to appear in Proceedings, 1987 IEEE International Conference on Robotics and Automation, Raleigh, NC, March 31 - April 2, 1987. Also, to appear as MIT VLSI Memo no. 87-369, March 1987.

Oded Z. Maimon and Stanley B. Gershwin, "Dynamic Scheduling and Routing for Flexible Manufacturing Systems that have Unreliable Machines," to appear in Proceedings, 1987 IEEE International Conference on Robotics and Automation, Raleigh, NC, March 31 - April 2, 1987. Also, to appear as MIT VLSI Memo no. 87-370, March 1987.

TALKS WITHOUT PROCEEDINGS

Stanley B. Gershwin, "Compensating for Uncertainties in Manufacturing Systems," Operations Research Symposium, Columbia University, NY, October 10, 1986.

M. L. Heytens, "Data Models for CAF," SRC Workshop on System Architecture for CIM, Berkeley, CA, November 13-14, 1986.

M. B. McIlrath, "Process Flow Languages," SRC Workshop on System Architecture for CIM, Berkeley, CA, November 13-14, 1986.

Duane S. Boning, "Wafer Profile Interchange Format," SRC Workshop on System Architecture for CIM, Berkeley, CA, November 13-14, 1986.

D. E. Troxel and R. Jayavant, "Instrumentation for Real-Time Control," SRC Workshop on System Architecture for CIM, Berkeley, CA, November 13-14, 1986.

P. Penfield, Jr., "Common CAF: What to Standardize?," SRC Workshop on System Architecture for CIM, Berkeley, CA, November 13-14, 1986.

Stanley B. Gershwin, "A Mathematical Framework for VLSI Manufacturing Systems Scheduling," Fall 1986 VLSI Research Review, MIT Microsystems Research Center, Cambridge, MA, December 15, 1986.

Stanley B. Gershwin, "Headaches, or How Manufacturing Systems Differ from Other Kinds of Queuing Networks," Conference on Networks of Queues and Their

Applications, sponsored by ORSA Technical Section/TIMS College on Applied Probability and cosponsored by SIAM, Rutgers University, New Brunswick, NJ, January 7-9, 1987.

Stanley B. Gershwin, "A Hierarchical Framework for Planning and Scheduling Manufacturing Systems," Boston University Colloquium on Recent Advances in Manufacturing Systems, Boston, MA, January 30, 1987.

P. Penfield, Jr., "Computer-Aided Fabrication of Integrated Circuits," invited talk, Conference on Advanced Research in VLSI, Stanford University, Palo Alto, CA, March 23-25, 1987.

A VISCOELASTIC BEM FOR MODELING OXIDATION

Thye-Lai Tung, Jerome Connor, and Dimitri A. Antoniadis
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

Abstract

A viscoelastic boundary element method has been developed to model the motion of silicon dioxide and silicon nitride during thermal oxidation of silicon. This technique uses Kelvin's solution reformulated according to the correspondence principle on viscoelasticity. Constant-velocity loading is chosen to ensure smooth variations in displacement and stress behavior for a wide range of relaxation times.

1 Introduction

A major problem in modeling thermal oxidation of silicon is the moving boundaries. Driven by the conversion of silicon to silicon dioxide of increased volume, boundaries change shape drastically during a local oxidation of silicon (LOCOS) process step. Conventional methods such as the finite element method (FEM) require a mesh to subdivide the simulation domain. To apply such methods to thermal oxidation, one must first develop computer codes to regenerate the mesh automatically and optimally as the oxide changes shape at every time step [1,2,3]. After that, one may have to deal with the transfer of stress history from the old mesh to the new one, depending on what oxide flow model is used. This is another difficult problem if one wishes to minimize numerical diffusion of stress distribution due to regriding.

In contrast to the FEM, the boundary element method (BEM) does not require a mesh in general since it does all calculations on the boundary but not in the interior. Because segments need not be regenerated, keeping track of stress history on the boundary is straight forward. However, previous efforts on applying the BEM to model oxide motion as viscoelastic flow suffered from some drawbacks. In Matsumoto's formulation [4], domain calculations were needed. Although restrictions on the grid were not as stringent as those in the FEM, excessive computation time was evident. Simulation setup might still be difficult for complex structures. Isomae attacked the problem differently by using a Laplace transform technique [5]. According to the correspondence principle of linear viscoelasticity, we may solve a viscoelastic problem through an equivalent elastostatic system in the Laplace transform space. Boundary element techniques for elastostatics do not require domain calculations, but unfortunately it is impossible to solve Laplace transforms analytically in any practical simulation situations. Numerical Laplace transform, as used

by Isomae, is deemed wasteful because a new solution must be generated at every time step.

In this paper we describe a generalized BEM for two-dimensional linear viscoelastic flow, with application to thermal oxidation. Instead of performing the Laplace transform on the global solution, we operate it on the fundamental equations. Our kernel functions are derived from Kelvin's solution used in elastostatic BEM applications [6]. A suitable excitation function is added to produce a time-varying body force that causes the load point to move with a constant velocity in a viscoelastic medium. This approach eliminates domain calculations and numerical Laplace transforms. Yet, it is as easy to use as those for elastostatics. The main disadvantage is that boundary conditions cannot be satisfied for all times, so the system must be solved periodically. But that is not a problem for thermal oxidation because a new solution must be obtained anyway for every time step. This formulation can handle all possible values of Poisson's ratio and a wide range of stress relaxation times, essentially encompassing viscous incompressible flow and elastic deformation.

2 Numerical Formulation

We implement the viscoelastic BEM using the so-called indirect formulation, which is also known as the classical or source method. Here we attempt to model a problem by putting "sources" on the boundary and adjust their strength so that the fields they generate match the prescribed boundary conditions. For a potential problem, the sources are simply electrical charges. The indirect formulation has an advantage over the direct formulation in that different components within the stress tensor or displacement vector can be obtained more readily.

The thermal oxidation process actually consists of two tightly coupled processes, namely oxidant diffusion, and oxide motion. For details on the oxidant diffusion process, boundary conditions, and numerical implementation of the integral equations, readers are referred to our previous paper on the same subject [7]. In that paper we use an incompressible viscous flow model for oxide motion; here we have a generalized model applicable to the oxide, silicon nitride and silicon substrate.

The two sets of integral equations for viscoelasticity are shown below:

$$\sigma_{ij}(\vec{x}) = \int_{\Gamma} \rho_k(\vec{\xi}) \sigma_{ijk}^*(\vec{x} - \vec{\xi}) d\Gamma$$

$$u_i(\vec{x}) = \int_{\Gamma} \rho_j(\vec{\xi}) u_{ij}^*(\vec{x} - \vec{\xi}) d\Gamma$$

where σ_{ij} , u_j , and ρ are the stress tensor, displacement vector, and source density respectively. Γ denotes the boundary. σ_{ijk}^* and u_{ij}^* are the kernels; their actual forms are given in the appendix. For compactness, Einstein notation has been used. To get the surface traction, we apply the following formula:

$$p_i(\vec{x}) = \sigma_{ji}(\vec{x})n_j(\vec{x})$$

where n_j is the direction cosine of the surface.

If we want to treat silicon nitride as an elastic material with stiffness, we have to modify the boundary condition of the oxide-nitride interface from a free surface condition $\vec{p} = 0$ to the following:

1. displacement vector is continuous across the interface,
2. surface tractions of the two materials are equal and opposite.

To ensure stability, the equations for silicon nitride layer and oxide region must be solved simultaneously. The resulting system matrix is larger but banded.

Likewise, we use the same approach if we desire to model the silicon substrate as an elastic foundation, instead of treating it as a rigid body. Note that due to the unique formulation of BEM, boundary conditions at $y = -\infty$ need not be specified for the silicon substrate.

In viscoelasticity, we must include past stress history in the present time step. Consider p_n^m , the normal component of the surface traction at time step m . The two different relaxation terms, as defined in the appendix, are kept separated and updated in the following way:

$$\begin{aligned} p_{n\alpha}^m &= p_{n\alpha} + p_{n\alpha}^{m-1} \exp(-\Delta t^m / \tau_\alpha) \\ p_{n\beta}^m &= p_{n\beta} + p_{n\beta}^{m-1} \exp(-\Delta t^m / \tau_\beta) \end{aligned}$$

where Δt^m is the time step size, τ_α and τ_β the characteristic relaxation time constants.

3 Simulation Results

We will demonstrate how stress is relieved via viscoelastic flow, the effect of the silicon nitride layer on the oxide shape and the stress distribution. The relaxation time, or equivalently the viscosity, of oxide has not been determined accurately. It is difficult to do so because, for a given temperature, the viscosity depends on the quality of the oxide, water contents, and the presence of other impurities. Most of the recent data on oxide

viscosity are inferred rather than measured directly in experiments, but they are useful as ballpark figures. In any case, it is a good exercise to vary the relaxation time to see how it affects the stress distribution.

Shown in Fig. 1 is the outline of a semi-recessed LOCOS structure plotted for every time step. In this simulation window of $1.6\mu\text{m}$ wide, the silicon nitride mask extends from $x = 0$ to $x = 0.96\mu\text{m}$ and is assumed to be totally flexible. Initially the structure has a pad oxide thickness of 200\AA . Oxidation is carried out at 925°C in a wet ambient for 3.4 hours to get a final field oxide thickness of 5000\AA . The Young's modulus and Poisson's ratio are taken to be $8 \times 10^{11} \text{ dynes}\cdot\text{cm}^{-2}$ and 0.194. Shown in Fig. 2a, 2b and 2c is the normal surface traction at the oxide-silicon interface corresponding to relaxation times ($\frac{\eta}{G}$) of 100 hours, 1 hour, and 1 minute respectively. The normal component of the surface traction is plotted for every time step, just like the outline of the oxide in Fig. 1. (Because the oxide shape is almost identical for all relaxation times, only one is shown in Fig. 1.) In all the 3 stress plots, there are two peaks in the compressible stress region (negative value range). The early peak occurs at the edge of the nitride mask ($x = 0.96\mu\text{m}$). This peak is due to the highly nonuniform oxidation rate in that region. As time progresses, the peak shifts to the left, further into the nitride mask, and gives rise to a late peak. As expected, stress decreases as the viscosity gets lower.

In the second part, we repeat the same simulations with the silicon nitride layer modeled as an elastic material. The Young's modulus and the Poisson's ratio of silicon nitride are assumed to be $3.29 \times 10^{11} \text{ dynes}\cdot\text{cm}^{-2}$ and 0.266 respectively [6]. The thickness of the nitride layer is $0.1\mu\text{m}$. The final shapes of the oxide and the nitride layer are shown in Fig. 3a, 3b, and 3c. As we can see, the nitride layer bends less as the oxide becomes less viscous and flows more readily. The corresponding values for peak stress are 2×10^{11} , 1.4×10^{11} , and $4.6 \times 10^{10} \text{ dynes}\cdot\text{cm}^{-2}$ respectively. The first two values are unrealistically high because they are the same order of magnitude as the elastic moduli of oxide. We find that in general we need those stress values in order to bend the nitride mask to a degree comparable to what is found in experiments. To keep stress down to a realistic level, silicon nitride must deform elastoplastically or viscoelastically. Unfortunately we don't have data on those behaviors.

4 Summary

A boundary element technique has been developed to model thermal oxidation of silicon in two dimensions. It can handle a wide range of relaxation times for the viscoelastic flow of oxide. Simulations of some simple structures have been demonstrated. Preliminary results indicate that it is inadequate to treat silicon nitride as an elastic material. A comprehensive characterization of silicon nitride is clearly needed.

5 Acknowledgment

The authors would like to thank J. Hui and P. Sutardja for help discussions. This worked is supported by DARPA and MCC.

Appendix

The fundamental solutions for viscoelastic flow induced by constant-velocity loading are given below. Note that the subscripts of ϕ denote partial derivatives.

$$\begin{aligned}\sigma_{111}^* &= [-K_\alpha(2y\phi_{xy} + 3\phi_x) - K_\beta(y\phi_{xy} + 5\phi_x)]E \\ \sigma_{121}^* &= [K_\alpha(2y\phi_{xz} - \phi_y) + K_\beta(y\phi_{xz} - 4\phi_y)]E \\ \sigma_{221}^* &= [K_\alpha(2y\phi_{xy} - \phi_x) + K_\beta(y\phi_{xy} + 3\phi_x)]E \\ u_{11}^* &= -7K_\alpha(1 - 2\nu)y\phi_y - (y\phi_y + \phi)\Delta t \\ u_{21}^* &= 7K_\alpha(1 - 2\nu)y\phi_x + y\phi_x\Delta t\end{aligned}$$

$$\begin{aligned}\sigma_{112}^* &= [K_\alpha(2y\phi_{xz} - 3\phi_y) + K_\beta(y\phi_{xz} + 2\phi_y)]E \\ \sigma_{122}^* &= [K_\alpha(2y\phi_{xy} + \phi_x) + K_\beta(y\phi_{xy} - 3\phi_x)]E \\ \sigma_{222}^* &= [K_\alpha(2y\phi_{xz} + \phi_y) + K_\beta(y\phi_{xz} + 4\phi_y)]E \\ u_{12}^* &= 7K_\alpha(1 - 2\nu)y\phi_x + y\phi_x\Delta t \\ u_{22}^* &= 7K_\alpha(1 - 2\nu)y\phi_y + (y\phi_y - \phi)\Delta t\end{aligned}$$

and

$$\begin{aligned}\phi(\bar{r}) &= \frac{1}{2} \log[2(\cosh(y) - \sin(x))] \\ K_\alpha &= \frac{6}{7} \frac{\eta}{E} [1 - \exp(-\frac{\Delta t}{\tau_\alpha})] \\ K_\beta &= \frac{2}{7} \frac{\eta}{E} [1 - \exp(-\frac{\Delta t}{\tau_\beta})] \\ \tau_\alpha &= \frac{3(3 - 4\nu)\eta}{E} \\ \tau_\beta &= \frac{2(1 + \nu)\eta}{E} \\ &= \frac{\eta}{G}\end{aligned}$$

where E is the Young's modulus, G the shear modulus, and ν the Poisson's ratio. The fundamental solutions are functions of time; they are evaluated at the end of a time step of size Δt . K_α and K_β decay exponentially with a time constant of τ_α and τ_β respectively when the load is removed.

References

- [1] A. Poncet, "Finite-Element Simulation of Local Oxidation of Silicon," *IEEE Trans. CAD*, Vol. CAD-4, p. 41, 1985.
- [2] P. Sutardja, Y. Shacham-Diamand, and W. Oldham, "Simulation of Stress Effects on the Reaction Kinetics and Oxidant Diffusion in Silicon Oxidation," presented at the IEDM, Dec. 1986, Los Angeles.
- [3] C. S. Rafferty and R. W. Dutton, "Modeling Corner Oxidation," presented at the NUPAD Workshop, Nov. 1986, Santa Clara.
- [4] H. Matsumoto and M. Fukuma, "Numerical Modeling of Nonuniform Si Thermal Oxidation," *IEEE Trans. Elec. Dev.*, Vol. ED-32, p. 132, 1985.
- [5] S. Isomae, S. Yamamoto, S. Aoki, and A. Yajima, "Oxidation-Induced Stress in a LOCOS Structure," *IEEE Elec. Dev. Lett.*, Vol. EDL-7, p. 368, 1986.
- [6] C. Brebbia, J. Telles, and L. Wrobel, *Boundary Element Techniques*, New York: Springer-Verlag, 1984.
- [7] T. Tung and D. Antoniadis, "A Boundary Integral Equation Approach to Oxidation Modeling," *IEEE Trans. Elec. Dev.*, Vol. ED-32, p. 1954, 1985.

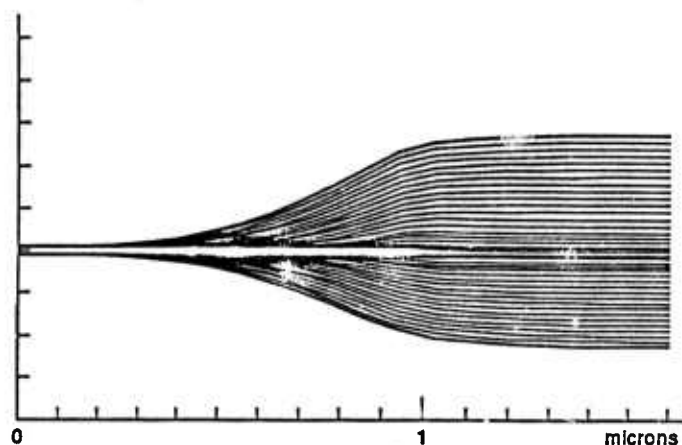
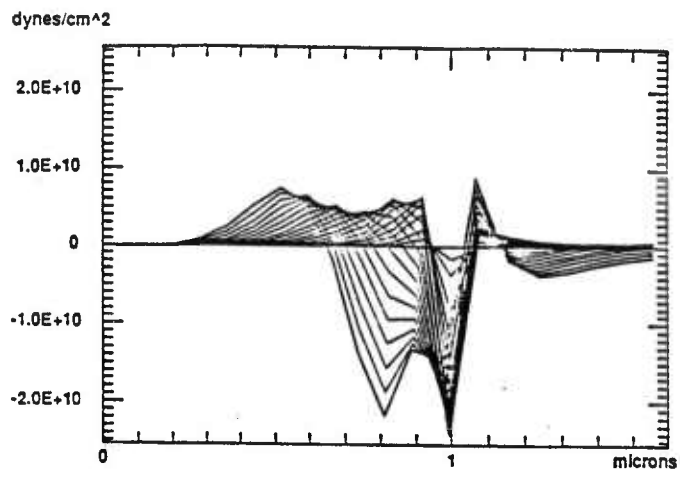
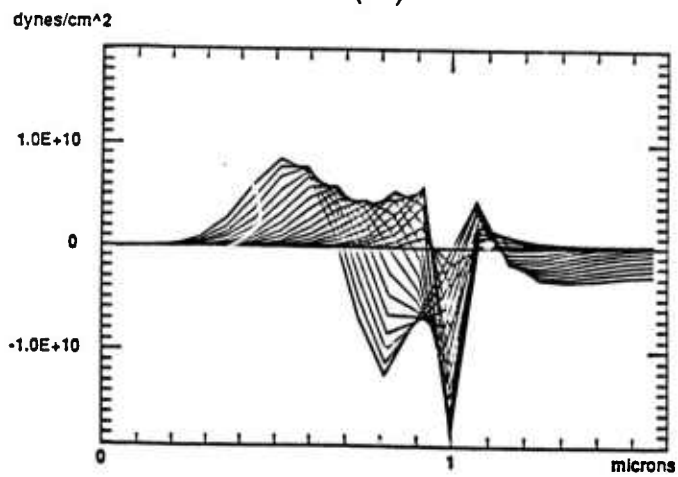


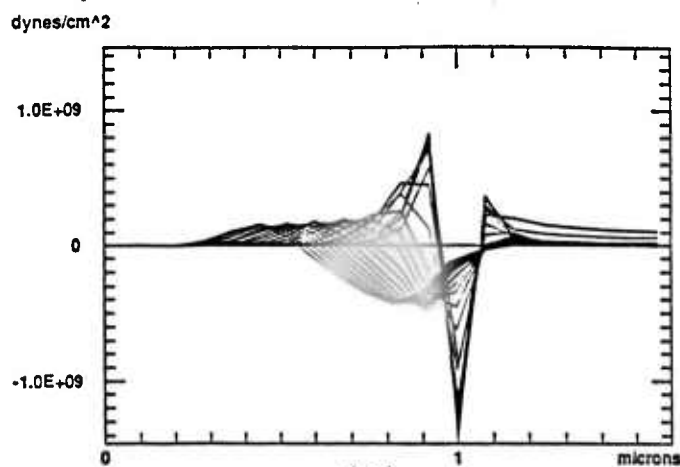
Fig. 1. Outline of oxide for every time step. Final field oxide thickness is $0.5\mu\text{m}$.



(2a)

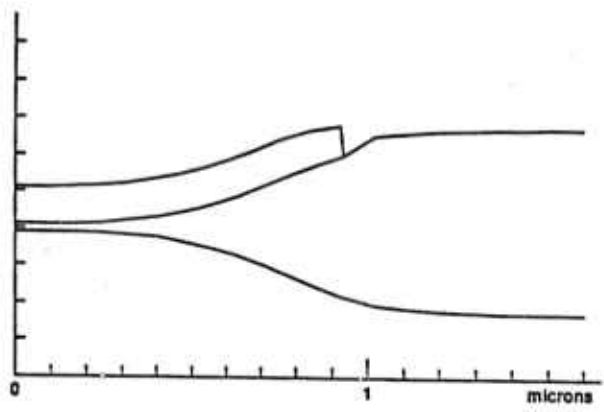


(2b)

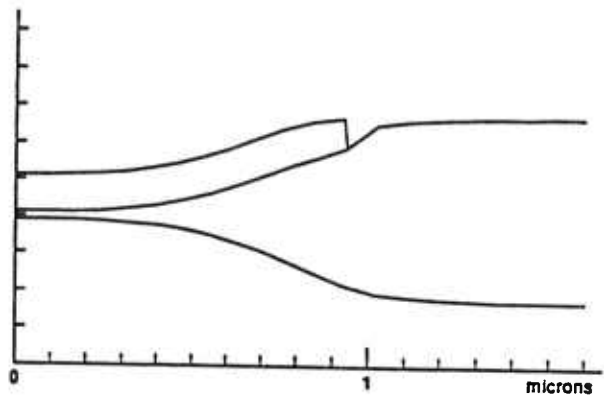


(2c)

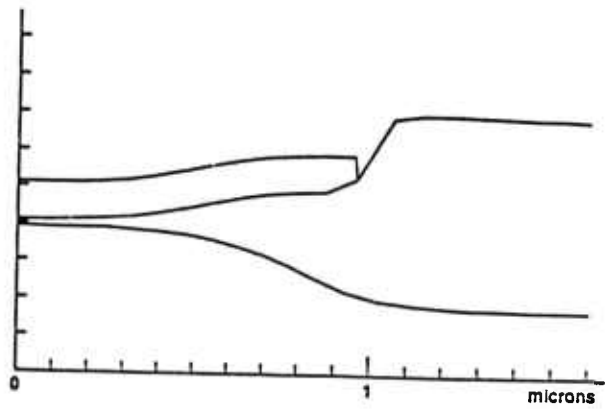
Fig 2. Normal surface traction at the oxide-interface. (a) $\tau_{\beta} = 100$ hours, (b) 1 hour, (c) 1 minute.



(a)



(b)



(c)

Fig 3. Effect of $0.1\mu\text{m}$ silicon nitride on oxide shape. (a) $\tau_{\beta} = 100$ hours, (b) 1 hour, (c) 1 minute.

SCHEDULING JOB SHOPS WITH DELAYS

Sheldon X.C. Lou
Garrett Van Ryzin
Stanley B. Gershwin

Massachusetts Institute of Technology

Abstract

In this paper, the presence of delay in a job shop is addressed. We show that delay is an important consideration in many manufacturing systems that are modeled as continuous flow processes. A scheduling policy for a job shop with delays is then derived using theoretical arguments and heuristics.

1 INTRODUCTION

It is well known that the optimal solution of the job shop scheduling problem is, in general, NP-hard [2]. Except for a few problems under very specific conditions, no computationally tractable solution for optimization can be found. Due to this formidable computational complexity, which necessitates the use of static, oversimplified models, traditional job shop scheduling approaches have not proven satisfactory in practice.

The approach proposed in [1], which in turn is a natural extension of [3], makes use of a hierarchical control structure to remedy these problems. A high level controller, similar to what described in [3], works at long time scales and deals only with work stations (work centers). It treats the production process as a continuous material flow. Its objective is to control the flow over a long time horizon so that the demand is satisfied as closely as possible and inventories are kept low, while keeping the system within production rate capacity constraints.

The actual loading of individual parts into machines is left to low level controllers which work at shorter time scale. The low level deals only with single work stations which have far fewer machines than the whole job shop. The low level attempts to fulfill the production goal determined by the high level controller. In this way, the two level controller can avoid the formidable computation requirements encountered in traditional approaches. Further, it dynamically adjusts the production to cope with real-time events.

While the two-level, continuous flow model does simplify the job shop scheduling problem, it comes with a hidden cost, namely that the differential equations representing the system must often include delay. To see this, notice that any work station that typically processes many parts at a time (i.e. where the number of total parts in processes is much greater than 1) will have average interarrival times that are much less than the processing time for a single part. For such a system, the time parts spend in the system cannot be ignored and thus delay must be explicitly included in the formulation.

In this paper, which is a summary of the work to appear in [8], we analyze the high level controller for systems with delay. In Section 2 we look at some examples of manufacturing systems with delay and show that a rather large class of manufacturing systems require delay formulations. In Section 3 we then show that a delay system can be approximated by a system of first order differential equations without delay. We use the results of [6] and let our approximation

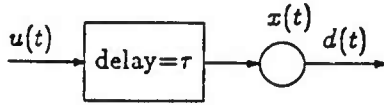


Figure 1: A single work station with delay

get arbitrarily good to arrive at a solution for the optimal control. Due to the difficulty of computing the optimal value function, we next explore a suboptimal strategy based on quadratic approximations to the value function. Finally, conclusions are presented in Section 4.

2 The Importance of Delay in Manufacturing Systems

We mentioned that delay arises in manufacturing systems that work on many parts at one time. We will now examine this phenomenon more closely and also try to indicate in what ways delay introduces difficulties into the scheduling problem.

Firstly, let us point out that introducing delay does not necessarily complicate the control problem. Consider, for example, a single work station with delay as shown in Fig. 1. where $x(t)$ is the inventory in the buffer, τ is the delay (processing time), $u(t)$ is the loading rate, which is bounded and the bound itself is a random variable ([3]), $d(t)$ is the demand rate, which is assumed to be deterministic and known. The dynamics of this system can be modeled as

$$\dot{x}(t) = u(t - \tau) - d(t) \quad (1)$$

By simply defining $\tilde{x}(t) = x(t + \tau)$ and $\tilde{d}(t) = d(t + \tau)$, both of which can be determined completely at time t , we can see this problem is no different than the non-delay problem. We simply use $x(t + \tau)$, rather than $x(t)$, as the current state and solve the problem as though there were no delay.

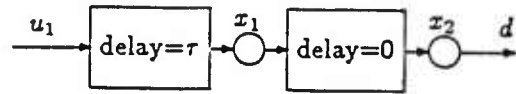


Figure 2: A two stage system

Unfortunately, delay cannot always be handled so simply. For example, consider the simple two stage system depicted in Fig. 2.

The system is described by

$$\dot{x}_1(t) = u_1(t - \tau_1) - u_2(t) \quad (2)$$

$$\dot{x}_2(t) = u_2(t) - d(t) \quad (3)$$

$$0 \leq x_1(t) \quad (4)$$

$$0 \leq u_1(t) \leq \alpha_1(t) \quad (5)$$

$$0 \leq u_2(t) \leq \alpha_2(t) \quad (6)$$

Suppose the constraints for u_1 and u_2 depend on some random processes (e.g. the machine state). We must determine u_1 and u_2 based on the *present* constraints yet the value of the future inventory, $x_1(t)$, depends on both the present $u_1(t)$ and the future $u_2(t)$, the constraints on which we do not know.

Another example where delay makes the problem more complex is in the scheduling of a reentrant job shop. A reentrant job shop is one where parts visit the same work station several times [7]. A simple reentrant job shop is shown in Fig. 3.

New parts are processed by the work station then go back to the same work station for a second process. After the second process is finished, they leave the system. There are buffers after the first and second processes whose levels are denoted by x_1 and x_2 respectively. Suppose the processing time for the second process is negligible. We then get the same system equations, (2) and (3). The only difference now is that the constraints on u_1 and u_2 are also coupled, namely,

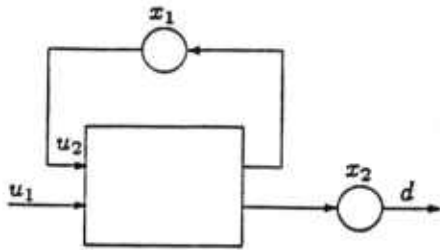


Figure 3: A reentrant job shop

$a_1 u_1(t) + a_2 u_2(t) \leq \alpha(t)$ for some a_1, a_2 and α . This further complicates the control. Thus, a single reentrant work station with delay also cannot be trivially handled.

In the next section, we expand on the ideas suggested by these examples and define the control problem in exact terms.

3 Solution For Delay Systems

To demonstrate our solution technique more clearly, we first investigate the simple problem described in (2) to (6). The technique, however, is extendible to more complex systems.

The objective functional is

$$\min_{u \in \Omega(\alpha)} \int g(x_1, x_2) dt \quad (7)$$

here $\Omega(\alpha)$ is a polyhedron defined by (5) and (6) and $g(\cdot)$ is some function of x_1 and x_2 . Without delay, this is the same formulation as in [3].

At time t , the parts in the first process that were loaded between $t - \tau_1$ and t will contribute to the future inventory and, therefore, should become part of the current state. Unfortunately, the problem then becomes an infinite dimensional one.

In order to overcome this difficulty we approximate the past u_1 by a finite dimensional first order system. We then let the approximation become better and better so that it approaches the original system.

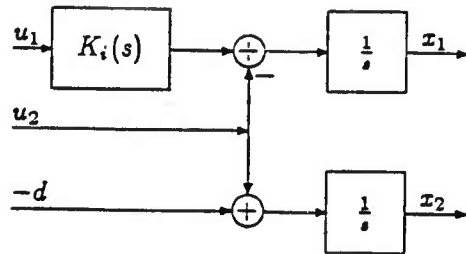


Figure 4: Diagram of two systems

Let us first define new variables $y_1(t)$ to $y_m(t)$ through the following equations.

$$\begin{aligned} \frac{\tau}{m} \dot{y}_1(t) &= u_1(t) - y_1(t) \\ \frac{\tau}{m} \dot{y}_2(t) &= y_1(t) - y_2(t) \end{aligned} \quad (8)$$

$$\vdots$$

$$\frac{\tau}{m} \dot{y}_m(t) = y_{m-1}(t) - y_m(t)$$

The initial conditions are set to zero at $-\infty$ and we assume that $u_1(-\infty) = 0$. Eq. (8) defines a cascade of m first order systems with time constant $\frac{\tau}{m}$. Its input and output are $u_1(t)$ and $y_m(t)$ respectively. As a motivation for using (8), note that its transfer function is $1/(1 + s\tau/m)^m$ which yields the well known limit $e^{-s\tau}$, the transfer function of a delay τ , as $m \rightarrow \infty$.

Now define

$$\dot{x}_1(t) = y_m(t) - u_2(t) \quad (9)$$

$$\dot{x}_2(t) = u_2(t) - d(t) \quad (10)$$

Combine (8)-(10), we obtain a new system. The diagrams of this system or the original system defined by (2) and (3) can be drawn as in Fig. 4.

The only difference between the system defined by (2) and (3) and the system defined by (8)-(10) is the first box $K_i(s)$. For the first system, it is a delay element with delay τ . For the second system, it is a linear system defined by (3). If we can show that for the same u_1, u_2

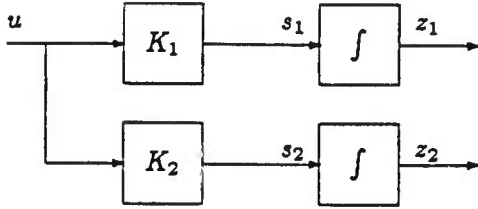


Figure 5: Compare two integrals

and d , the output of the first system approaches the output of the second one, then we can establish an equivalence between the two systems. By superposition, it is sufficient to show that the integral of $y_m(t)$ approaches the integral of $u_1(t - \tau)$ as m goes to infinity.

To show this result, we compare the two systems shown in Fig. 5. In Fig. 5, $K_1(s)$ is the system defined by (8) and $K_2(s)$ is a pure delay of τ . We will prove that $z_1(t)$ approaches $z_2(t)$ uniformly in t as $m \rightarrow \infty$. First we need the following lemma which is similar to [6].

Lemma 1 *If $u(t)$ is differentiable with $|\dot{u}(t)| < K$ for all $t \in (-\infty, +\infty)$, then*

$$\lim_{m \rightarrow \infty} \sup_{t \in (-\infty, +\infty)} \|s_1(t) - s_2(t)\| = 0$$

Proof: (see [8]).

The integrals in Fig. 5 start from $-\infty$. Since the initial conditions are zero at $-\infty$ and $u(-\infty)$ equals zero, we can switch the linear operators K_1, K_2 with the integrators, $1/s$. Because $u(t)$ is bounded, the integral of $u(t)$ has a bounded derivative. Combining these facts with Lemma 1, we obtain the following lemma.

Lemma 2 *If $u(t) \in L_1$ and is bounded, then*

$$\lim_{m \rightarrow \infty} \sup_{t \in (-\infty, +\infty)} \|z_1(t) - z_2(t)\| = 0$$

Proof: (see [8]).

Using Lemma 2, we see that the output of the system defined by (8)-(10) approaches the output of the system defined by (2) and (3). Therefore, if u_1 is optimal for the first system, it will also be optimal for the second one.

We will now consider the optimal control for the system defined by (8)-(10). Define $x = [x_1 \ x_2 \ y_1 \ \dots \ y_m]'$, $u = [u_1 \ u_2]'$. This system can be written in a compact form as

$$\dot{x} = Ax + Bu + Cd \quad (11)$$

Using the same approach as in [3], it can be shown that the optimal control u for the problem defined by (11) and the constraints (4)-(6) can be obtained by solving

$$\min_{u \in \Omega(\alpha)} \nabla_x J^*(x, \alpha) Bu \quad (12)$$

Eq. (12) can be rewritten as

$$\min_{u \in \Omega(\alpha)} \left[\frac{\partial J^*}{\partial x_1} u_2 - \frac{\partial J^*}{\partial x_2} u_1 + \frac{\partial J^*}{\partial y_1} u_1 \right] \quad (13)$$

Where $J^*(x, \alpha)$ is the *optimal* cost to go. Unfortunately, J^* remains unknown and is, even for simple problems, difficult to compute. Therefore, we seek an approximation for J^* . Experience ([4],[5]) shows that satisfactory results can be obtained with relatively crude approximations for J^* . The one we will use is in a quadratic form with coefficients that are functions of α , namely,

$$J^*(x, \alpha) \cong x' R(\alpha) x + S(\alpha) x \quad (14)$$

Then

$$\nabla_x J^*(x, \alpha) \cong R(\alpha) x + S(\alpha) \quad (15)$$

Using (15) we can rewrite (13) as

$$\min_{u \in \Omega(\alpha)} [\beta_1(x, \alpha) u_1 + \beta_2(x, \alpha) u_2] \quad (16)$$

where

$$\beta_i(x, \alpha) = \sum_{j=1}^m p_{ij}(\alpha) y_j + \sum_{j=1}^2 q_{ij}(\alpha) x_j + \rho_j(\alpha) \quad (17)$$

Letting m go to infinity we get

$$\beta_i = \int_0^{\tau} f_i(\sigma, \alpha) u_1(t-\sigma) d\sigma + \sum_{j=1}^2 q_{ij}(\alpha) x_j + \rho_j(\alpha) \quad (18)$$

Eq. (16) and (18) describe the sub-optimal control law for our system. Note that instead of a very complex dynamic program, the control (production rates) is determined by a linear program (16). The problem is further simplified due to the simple structure of $\Omega(\alpha)$. This calculation can easily be performed in real time.

The terms in β_i are easy to interpret. Note that β_i is a function of x_i , α and the past control u_1 . The second and third terms in β_i are identical to the terms found in the quadratic approximation presented in [3] for a non-delay system. Added to this is a convolution of the control $u_1(t)$ with some weighting function $f_i(\sigma)$. If a constant weighting is applied, the first term would be the integral of $u_1(t)$ from $t - \tau$ to t , which is the number of the parts currently under processing. Other weighting functions are, of course, possible.

Further research will be conducted to obtain the correct form of β_i . Further, since a real manufacturing system is much more complex than the system described here, more complex system models will be investigated which will take into account factors such as reentrant process and time varying demand. Finally, numerical computations and simulation experiments will be used to help develop solution techniques.

4 Conclusion

In this paper, we examined the effects of delay in manufacturing systems. We proposed a general model for a network of work stations that includes delay. We then presented a technique for analyzing delay systems by augmenting the states to include an approximation of the past control. Using quadratic approximations to the optimal value function, we show that the control takes a particularly simple form.

REFERENCES

1. S.X.C. Lou, *Job Shop Scheduling Using Flow Rate Control*, M.I.T. Laboratory for Information and Decision Systems, LIDS-P-1618, September, 1986.
2. R.W. Conway, W.L. Maxwell, *Theory of Scheduling*, Addison-Wesley, Reading, Mass., 1967.
3. J. Kimemia, S.B. Gershwin, *An Algorithm for the Computer Control of a Flexible Manufacturing System*, IIE Transactions, Vol. 15, No.4, December 1983, pp. 353-362.
4. S.B. Gershwin, R. Akella, and Y.F. Choong (1985), *Short-Term Production Scheduling of an Automated Manufacturing Facility*, IBM Journal of Research and Development, Vol. 29, No. 4, pp 392-400, July, 1985.
5. R. Akella, Y.F. Choong, and S.B. Gershwin (1984), *Performance of Hierarchical Production Scheduling Policy*, IEEE Transactions on Components, Hybrids, and Manufacturing Technology, Vol. CHMT-7, No.3, September, 1984.
6. I.M. Repin, *On the Approximate Replacement of Systems with Lag by Ordinary Dynamical Systems*, Journal of Applied Mathematics and Mechanics, Vol. 29, 1965, pp. 254-264 (translation of PMM, Vol.29, No.2, 1965, pp.226-235).
7. S.C. Graves, H.C. Meal, D. Stefek, A.H. Zeghmi (1983), *Scheduling of Re-Entrant Flow Shops*, Journal of Operations Management, Vol.3, No.4, August 1983, pp 197-207.
8. G. Van Ryzin, *Control of Manufacturing Systems With Delays* M.S. Thesis under preparation, Laboratory for Information and Decision Systems, MIT.

Dynamic Scheduling and Routing For Flexible Manufacturing Systems That Have Unreliable Machines

by

Oded Z. Maimon[†] and

Stanley B. Gershwin^{*}

[†] Computer Integrated Manufacturing
Digital Equipment Corporation
Maynard, Massachusetts 07154

^{*} Laboratory for Information
and Decision Systems
Massachusetts Institute of Technology
77 Massachusetts Avenue
Cambridge, Massachusetts 02139

ABSTRACT

This paper presents a method for real-time scheduling and routing of material in a Flexible Manufacturing System (FMS). It extends the earlier scheduling work of Kimemia and Gershwin. The FMS model includes machines that fail at random times and stay down for random lengths of time. The new element is the capability of different machines to perform some of the same operations. The times that different machines require to perform the same operation may differ. This paper includes a model, its analysis, a real-time algorithm, and examples.

1. INTRODUCTION

Purpose

The purpose of this paper is to develop an algorithm to calculate real-time loading and routing decisions for a Flexible Manufacturing System (FMS). An algorithm for calculating loading decisions for such systems has been described in earlier papers (Kimemia and Gershwin, 1983; Gershwin, Akella, and Choong, 1985; Akella, Gershwin, and Choong, 1985). An algorithm for routing decisions is described in Maimon and Choong (1985). Here, routing and loading are calculated together.

As in the earlier papers, the problem is to decide which part should be dispatched next into a set of machines. These machines are capable of performing work on a set of different part types with no time lost for setting up. Decisions are made in response to disruptions of the operation of the system caused by machine failures, and according to the surplus or backlog for each part type. Whenever a machine changes state (i.e., fails or is repaired), a new schedule and a new routing scheme is calculated via a feedback law.

A limited form of routing flexibility was allowed in the earlier work. Only identical machines could perform the same operation. In that case, a part could be routed to the first available copy. The purpose of this paper is to deal with systems in which machines are not identical, but where different machines may perform some of the same operations. Different machines may therefore have overlapping capability, and different machines performing the same operation may take different lengths of time to do it. The routing problem is therefore to choose among alternate machines for some or all the operations. A model capable of analyzing such issues model is required for the study of certain real systems.

Kimemia and Gershwin (1983) proposed a routing algorithm to go along with their scheduling scheme. However, while the scheduling method was effective, the routing method was not. In particular, the routing decisions that would have been calculated by the method suggested there might not be feasible.

Examples

This work was motivated by two actual Flexible Manufacturing Systems, one from the electronics industry and one from the metal cutting industry.

We examined a robotic system for the assembly of printed circuit boards (PCB), particularly the part of the system where oddly-shaped components are inserted to the board (such as large electrolytic capacitors, switches, and connector strips). These components cannot be inserted with existing dedicated automated machines (e.g., SIP, DIP, VCD), because of their variability and special handling and assembly requirements. However, some types of robots (e.g., Adept, IBM 7575), equipped with appropriate fixtures and end effector tools, can meet the job require-

ments (e.g., tolerance better than 0.005"), and are adaptable (programmable) so that they can handle different types of odd components.

As a result, different operations (insertions of odd component types) can be performed by different robots, but the amount of time required for a given operation depends on the speed of the robot that performs it. Also, each robot has different configurations (e.g., tools) and inherent capabilities (e.g., accuracy and repeatability), which results in different subsets of operations that each robot can handle (with nonempty intersections among those subsets).

As a consequence, not only does the input rate of part types into the system have to be determined, but also the decision of where to send each part for each operation (among the possible alternative robots) has to be made.

Such systems are usually justified economically only if the production volume is quite high (e.g., hundreds of thousands of components inserted per year) and the variety is high. Because of their flexibility, they are expected to meet demands that vary in the short term and that require high utilization. The work presented here aims to improve system performance (e.g., to lead to higher throughput and reduced WIP while meeting production demands).

Another type of manufacturing system is comprised of conventional and advanced machining centers. The latter are capable of performing different operations, with varied capabilities. For example, some machining centers can do drilling and milling operations that otherwise require two different conventional machines. Also there are 3- and 5-axis machining centers. The latter can do more operations than the former without changing the part fixturing.

As in an electronic insertion system, the scheduling and routing problem in a system of several machining centers is not only to decide on the input flow rate of each part type, but also where each operation should be done among alternative machines with different capabilities.

Literature Survey

In this paper we present a method that considers, at the same time, two functions -- short-term scheduling and routing -- based on a global view of the system. Many references consider just one

of these functions. For example, Whitt (1986) presents a method which can be used just for the local routing decisions. Although his paper develops generic queueing methodology, we use his results to show an example of local routing considerations.

By local routing decisions we refer to a situation by which a customer (or a part) has to join one of several queues. These queues represent, for example, the input buffers to workstations. The alternative queues are those of the alternative workstations that can perform the next operation on a part, which has just finished a particular operation.

Whitt shows that in some cases, the system average delay is not always minimized by customers joining the queue that minimizes their own individual expected delay. This result suggests that decisions should be made only when taking a global view of the system.

Routing is treated in papers by Hahne (1981), Tsitsiklis (1981), and Seidmann and Schweitzer (1984). Hahne and Tsitsiklis deal with only two choices and machines whose randomness is due to failure and repair. Seidmann and Schweitzer have many choices, but the randomness is due to variations in processing times. In all cases, the full system is not considered. Instead, only one decision point is considered, and decisions are made on a purely local basis.

By contrast, we consider the whole system and do not treat local conditions in detail. This suggests that a hierarchical decision policy, in which both kinds of decisions -- local and global -- are made separately, may be appropriate. The local decisions should be made in a way that is consistent with the decisions made on a global basis.

Outline of Paper

Section 2 states the problem. Section 3 contains our solution, which is based on dynamic programming. Section 4 describes some numerical examples and simulation results. Conclusions and new research directions are discussed in Section 5.

2. PROBLEM STATEMENT

Section 1 describes two situations in which short-term scheduling and routing decisions are required. In this section we represent such manufacturing systems with a mathematical model.

The input to the problem is the production requirements and process data in the form of process plans and routing sheets. They specify the operations that each part type has to go through, together with a partial precedence relation among the operations. For each operation, a set of alternative machines, and the time for the operation at each machine, (and machine reliability) are specified.

We seek a feedback law which determines when each part should be released into the system and which route it should take when it enters. The release time and the route may be functions of the current repair state of each machine as well as the current production level of each part type.

Model

The FMS consists of M work stations, and work station m consists of L_m identically configured machines. A family of N part types is being produced. The production rate of part type n at time t is $u_n(t)$.

Let d_n be the demand rate for type n parts. This is a rate that is specified by higher level decision-makers in the decision hierarchy. We assume here that it is constant over the time interval of interest. The model is unchanged if it is deterministic but time-varying, but the computation is made more difficult. Requirements are often stated in terms of production required over some specified time interval; we convert this to demand rates.

Let $x_n(t)$ be the surplus (if positive) or backlog (if negative) of type n parts at time t . It is the difference between production and demand, and is given by

$$\frac{dx_n}{dt} = u_n(t) - d_n. \quad (1)$$

The states of the work stations are given by $\alpha_m(t)$. This is an integer which indicates the number of machines of work station m that are operational at time t . The vector α is assumed to be the state of a continuous time Markov process with rates λ , so that

$$\text{prob} [\alpha(t+\delta t) = b \mid \alpha(t) = a] = \lambda_{ab}. \quad (2)$$

Recall that different work stations may be available for some operations, and that they perform them at different speeds. Routing is the decision of which work station will perform each operation.

Let y_{nm}^k be the rate at which work station m performs operation k on type n parts. (Since only a few operations among all those that are possible are performed on each part type, most of these variables are 0.) The relationship between u_n and y_{nm}^k is given by

$$u_n = \sum_m y_{nm}^k \text{ for any } k \text{ and } n, \quad (3)$$

In this section, we formulate an optimization problem whose solution is the optimal set of y_{nm}^k variables as a function of time. In Section 3, we describe a suboptimal solution.

Capacity

The rate of flow of material into the system is limited by the rate at which machines can do operations. Each operation takes a finite amount of time, and no machine can be busy more than 100% of the time. A fundamental assumption is that there is no buffering inside the system. This reduces the total work in process, but increases the need for effective routing and scheduling.

Let τ_{nm}^k be the amount of time that a machine in work station m requires to do operation k on a part of type n . The rate at which machines of that station have to do such operations has already been defined as y_{nm}^k .

During a short interval of length T , the expected number of operations performed by the machines is $y_{nm}^k T$. (It is assumed that the interval is short so that no repairs or failures take place during it.) The total amount of time that all of the machines of station m are performing operation k on part type n is $y_{nm}^k \tau_{nm}^k T$. The expected total amount of time that the machines of station m are performing all operations on all part types is

$$\sum_n \sum_k y_{nm}^k \tau_{nm}^k T.$$

The total amount of time available on all the machines of station m is $\alpha_m T$ if α_m machines are operational. Therefore,

$$\sum_n \sum_k y_{nm}^k \tau_{nm}^k \leq \alpha_m.$$

To summarize, the y flow rates must satisfy the following set of equations and inequalities:

$$y_{nm}^k \geq 0 \quad \forall k, m, n \quad (4)$$

$$\sum_k \sum_n y_{nm}^k \leq \alpha_m \text{ for every machine } m. \quad (5)$$

$$\sum_m y_{nm}^k = \sum_m y_{nm}^{k_n} \text{ for all } k \neq k_n \text{ and all part types } n, \quad (6)$$

where k_n is the name of the first operation performed on parts of type n . Denote by $\Omega(\alpha)$ the set of all y flow rates that satisfy (4) - (6).

Note that $\Omega(\alpha)$ is a random set. As machines fail and are repaired the instantaneous capacity changes. The rates that material flows into the system must change as $\Omega(\alpha)$ changes, as well as in anticipation of these changes.

Cost Function

We seek a policy that minimizes a cost of the form

$$E \left[\int_0^T g(x(s)) ds \mid x(0) = x_0, \alpha(0) = \alpha_0 \right] \quad (7)$$

in which T is the short term period, such as an eight hour shift and $g(\cdot)$ is a positive convex function. We assume the cost function does not reflect true costs, but instead is chosen to lead to desirable behavior. Thus, the details of $g(\cdot)$ are not important. In Section 3 we describe an approximation method which uses only certain features of the cost function.

Dynamic Programming Formulation

The optimization problem can be written:

$$\text{minimize } J(x_0, \alpha_0, 0)$$

subject to dynamics given by (1) and (2) and $y \in \Omega(\alpha)$.

Comparison with Kimemia and Gershwin

Kimemia and Gershwin (1983) formulated an optimization problem in terms of u of equation (3). This formulation is correct when there are no route choices except among identical machines. However, they assumed that they could ignore (6) even when route choice existed, and then determine y from u after solving the problem. This assumption is not correct; the above formulation is. Without (6), the choice of routes achieved may not be feasible, and (3) would not necessarily hold.

3. SOLUTION

Following the usual dynamic programming practice, define

$$J(x, \alpha, t) =$$

$$\min_{y \in \Omega(\alpha)} E \left[\int_t^T g(x(s)) ds \mid x(t) = x, \alpha(t) = \alpha \right]. \quad (8)$$

This function satisfies the Bellman equation (Bertsekas, 1976), which takes the following form:

$$0 = \min_{y \in \Omega(\alpha)} \left\{ g(x(t)) + \sum_n \frac{\partial J}{\partial x_n} \left(\sum_m y_{nm}^{k_n} - d_n \right) + \frac{\partial J}{\partial t} + \sum_{\lambda, \beta} \lambda_{\alpha\beta} J(x, \beta, t) \right\}. \quad (9)$$

This equation has the following interpretation: we seek a function $J(x, \alpha, t)$ such that the values of $y(x, \alpha, t) \in \Omega(\alpha(t))$ that minimize the right hand side of (9) cause that expression to be zero. This is a nonlinear partial differential equation which we cannot expect to have an analytic solution. (However, in the case of a single part type and a single machine, Akella and Kumar (1986) were able to find a closed form solution.)

If (9) has a solution, the optimal control y satisfies the following linear programming problem. Note that the cost coefficients are time-varying.

$$\left. \begin{array}{l} \min \sum_n \frac{\partial J}{\partial x_n} \left(\sum_m y_{nm}^{k_n} \right) \\ \text{subject to} \\ y \in \Omega(\alpha) \end{array} \right\} \quad (10)$$

It is important to recognize that this is a feedback control law since J and Ω are functions of x and α . The solution y is therefore a function of x and α .

Note that J is positive since it is the expected value of the integral of g , a positive quantity. Note also that feedback law (10) minimizes

$$\frac{dJ}{dt} = \sum_n \frac{\partial J}{\partial x_n} \left(\sum_m y_{nm}^{k_n} - d_n \right) + \frac{\partial J}{\partial t} \quad (11)$$

while α is constant. This is because y appears in (9) only in the same term in which it appears in (11). If α remains constant long enough, and there is a $y \in \Omega(\alpha)$ such that (11) is negative, then J

eventually reaches a minimum. We call the value of x that produces this minimum the hedging point and write it x_α^H . If possible the production rate should remain at a rate that keeps x at the hedging point. A positive hedging point serves as insurance for future disruptions.

After J reaches this minimum, J and x are both constant. Therefore, at the minimum,

$$\sum_m y_{nm}^{k_n} - d_n = 0 \quad (12)$$

and

$$\frac{\partial J}{\partial t}(x_\alpha^H, \alpha, t) = 0 \quad (13)$$

If there is no $y \in \Omega(\alpha)$ that satisfies (12), then J cannot reach a minimum for finite x . That is, the production lags behind the demand requirements and $x(t)$ decreases. This is because too many machines are currently down to allow production to equal demand.

There are reasons to believe that the solution of linear programming problem (10) provides a satisfactory scheduling and routing algorithm even if an approximate J function is used. This was the simulation experience reported by Gershwin, Akella, and Choong (1985) and Akella, Gershwin, and Choong (1985).

In addition, it is likely that the repair and failure processes are not actually exponential, not actually independent of the machine utilizations (as assumed in Section 2), and do not have the exact λ parameters that would be used in (9) if an exact solution could be calculated. Also, the g function does not necessarily represent true costs, but rather is chosen to obtain a desired behavior. For these reasons, it would be a mistake to work very hard to get an exact J .

Therefore, a reasonable strategy is to select a J function that has the correct qualitative properties and that is easy to calculate and work with. Such a function is positive and has a minimum at the hedging point (for every α such that the demand is feasible for that α). Gershwin, Akella, and Choong (1985) use a quadratic function,

$$J = \frac{1}{2}x^T A(\alpha)x + b(\alpha)^T x + c(\alpha).$$

Akella, Maimon, and Gershwin (1987) demonstrate a technique for calculating a set of values for $A(\alpha)$, $b(\alpha)$, and $c(\alpha)$, from a specified g , for a model similar to the one presented here.

4. EXAMPLES

Example 1: Three-Machine System

Consider a three-machine system that makes two part types. Machine 1 can do operations only on Type 1; Machine 2 can only work on Type 2; and Machine 3 can do operations on both. In fact, Machine 3 can do the same operations that Machines 1 and 2 can do. Thus Type 1 parts can go to Machine 1 or Machine 3 and Type 2 parts can go to Machine 2 or Machine 3. The problem is to decide where to send each of the parts and how frequently to send them into the system.

The capacity set $\Omega(\alpha)$ is given by:

$$\tau_{11}^1 y_{11}^1 \leq \alpha_1 \quad (14)$$

$$\tau_{22}^2 y_{22}^2 \leq \alpha_2 \quad (15)$$

$$\tau_{13}^1 y_{13}^1 + \tau_{23}^2 y_{23}^2 \leq \alpha_3 \quad (16)$$

$$y_{11}^1, y_{13}^1, y_{22}^2, y_{23}^2 \geq 0 \quad (17)$$

The production surplus and backlog dynamics are:

$$\dot{x}_1 = y_{11}^1 + y_{13}^1 - d_1 \quad (18)$$

$$\dot{x}_2 = y_{22}^2 + y_{23}^2 - d_2 \quad (19)$$

If $J(x, \alpha, t)$ is known, then the optimal routing and scheduling policy y satisfies

$$\min_{y \in \Omega(\alpha)} \frac{\partial J}{\partial x_1} (y_{11}^1 + y_{13}^1) + \frac{\partial J}{\partial x_2} (y_{22}^2 + y_{23}^2) \quad (20)$$

This is a feedback control law since the constraint set is a function of α and the partial derivatives are functions of x and α . To solve this linear programming problem, several cases must be considered. Figure 1 demonstrates the various regions of $\partial J / \partial x$ -space that have different solutions. The regions are indicated, as well as the values of y_{mn}^k that are optimal in those regions. Also indi-

cated is which of the following conditions that determine the values.

$$\begin{aligned} \frac{\partial J}{\partial x_1} > 0 \text{ (Regions I and III)} \\ \Rightarrow y_{11}^1 = 0, y_{13}^1 = 0 \end{aligned} \quad (A)$$

$$\begin{aligned} \frac{\partial J}{\partial x_2} > 0 \text{ (Regions I and II)} \\ \Rightarrow y_{22}^2 = 0, y_{23}^2 = 0 \end{aligned} \quad (B)$$

$$\begin{aligned} \frac{\partial J}{\partial x_1} < 0 \text{ (Regions II, IV, V, and VI)} \\ \Rightarrow y_{11}^1 = \frac{\alpha_1}{\tau_{11}} \end{aligned} \quad (C)$$

$$\begin{aligned} \frac{\partial J}{\partial x_2} < 0 \text{ (Regions III, IV, V, and VI)} \\ \Rightarrow y_{22}^2 = \frac{\alpha_2}{\tau_{22}} \end{aligned} \quad (D)$$

$$\begin{aligned} \frac{\partial J}{\partial x_1} < 0 \text{ and } \frac{\partial J}{\partial x_2} > 0 \text{ (Region II)} \\ \Rightarrow y_{13}^1 = \frac{\alpha_1}{\tau_{13}} \end{aligned} \quad (E)$$

$$\begin{aligned} \frac{\partial J}{\partial x_1} > 0 \text{ and } \frac{\partial J}{\partial x_2} < 0 \text{ (Region III)} \\ \Rightarrow y_{23}^2 = \frac{\alpha_3}{\tau_{23}} \end{aligned} \quad (F)$$

If both derivatives are negative (Regions IV and V), y_{11}^1 and y_{22}^2 are already determined. The remaining variables, y_{i3}^i ($i = 1, 2$), minimize

$$\frac{\partial J}{\partial x_1} y_{13}^1 + \frac{\partial J}{\partial x_2} y_{23}^2 \quad (21)$$

subject to (16). The solution is

$$\begin{aligned} \left\{ \frac{1}{\tau_{23}} \frac{\partial J}{\partial x_2} - \frac{1}{\tau_{13}} \frac{\partial J}{\partial x_1} \right\} < 0 \text{ (Region V)} \\ \Rightarrow y_{23}^2 = \frac{\alpha_3}{\tau_{23}} \text{ and } y_{13}^1 = 0 \end{aligned} \quad (G)$$

$$\begin{aligned} \left\{ \frac{1}{\tau_{23}} \frac{\partial J}{\partial x_2} - \frac{1}{\tau_{13}} \frac{\partial J}{\partial x_1} \right\} > 0 \text{ (Region IV)} \\ \Rightarrow y_{13}^1 = \frac{\alpha_3}{\tau_{13}} \text{ and } y_{23}^2 = 0. \end{aligned} \quad (H)$$

In each of these regions, the control y_{mn}^k moves the state x_n through the dynamics [(18) and (19)]. The state moves to a boundary and then to another region. However, there is one exception. In both Regions IV and V the state moves toward the common boundary, which is given by

$$\left\{ \frac{1}{\tau_{23}} \frac{\partial J}{\partial x_2} - \frac{1}{\tau_{13}} \frac{\partial J}{\partial x_1} \right\} = 0 \text{ (Region VI). (I)}$$

If we follow rules (G) and (H), the state will move back and forth across the boundary in an unrealistical manner. This is called *chattering*. It occurs because the problem is *singular*, and a remedy is suggested by Geršwin, Akella,

and Choong (1985). A strategy is found which, when x reaches Region VI, keeps x in Region VI. That is, it maintains (I). It does this by determining y_{mn}^k which minimizes (21) subject to (16) and

$$\frac{d}{dt} \left\{ \frac{1}{\tau_{23}} \frac{\partial J}{\partial x_2} - \frac{1}{\tau_{13}} \frac{\partial J}{\partial x_1} \right\} = 0. \quad (22)$$

This is simplified by assuming that J is quadratic:

$$J = \frac{1}{2} x^T A(\alpha) x + b(\alpha)^T x + c(\alpha). \quad (23)$$

Then

$$\frac{\partial J}{\partial x_1} = A_{11} x_1 + A_{12} x_2 + b_1 \quad (24)$$

and

$$\frac{\partial J}{\partial x_2} = A_{21} x_1 + A_{22} x_2 + b_2. \quad (25)$$

If

$$\frac{\partial J}{\partial x_1} < 0$$

and y is chosen so that

$$\left\{ \frac{1}{\tau_{23}} \frac{\partial J}{\partial x_2} - \frac{1}{\tau_{13}} \frac{\partial J}{\partial x_1} \right\} = 0, \quad (26)$$

then

$$\begin{aligned} \frac{1}{\tau_{23}} (A_{21} x_1 + A_{22} x_2 + b_2) \\ - \frac{1}{\tau_{13}} (A_{11} x_1 + A_{12} x_2 + b_1) = 0. \end{aligned} \quad (27)$$

Since this is true for more than just one instant, its first derivative with respect to t is also 0. That is,

$$\begin{aligned} \frac{1}{\tau_{23}} (A_{21} \dot{x}_1 + A_{22} \dot{x}_2 + b_2) \\ - \frac{1}{\tau_{13}} (A_{11} \dot{x}_1 + A_{12} \dot{x}_2 + b_1) = 0, \end{aligned} \quad (28)$$

or,

$$\begin{aligned} \frac{1}{\tau_{23}} [A_{21}(y_{11}^1 + y_{13}^1 - d_1) + A_{22}(y_{22}^2 + y_{23}^2 - d_2) + b_2] \\ - \frac{1}{\tau_{13}} [A_{11}(y_{11}^1 + y_{13}^1 - d_1) + A_{12}(y_{22}^2 + y_{23}^2 - d_2) + b_1] = 0, \end{aligned} \quad (29)$$

From (C) and (D),

$$\frac{1}{t_{23}^2} \left(A_{21} (\alpha_1 / \tau_{11}^1 + y_{13}^1 - d_1) + A_{22} (\alpha_2 / \tau_{22}^2 + y_{23}^2 - d_2) + b_2 \right) - \frac{1}{t_{13}^1} \left(A_{11} (\alpha_1 / \tau_{11}^1 + y_{13}^1 - d_1) + A_{12} (\alpha_2 / \tau_{22}^2 + y_{23}^2 - d_2) + b_1 \right) = 0. \quad (30)$$

Now (16) (as an equality) and (30) are two equations in two unknowns, y_{13}^1 and y_{23}^2 . The solution is

$$y_{13}^1 = \frac{\left[\frac{1}{t_{23}^2} \left(A_{21} \left(\frac{\alpha_1}{t_{11}^1} - d_1 \right) + \alpha_3 (A_{22} + A_{12}) \right) + \frac{1}{t_{22}^2} \alpha_2 (A_{22} + A_{12}) \right] + \frac{A_{11}}{t_{13}^1} \left(d_1 - \frac{\alpha_1}{t_{11}^1} \right) - d_2 (A_{12} + A_{22}) + b_1 + b_2}{\left[\frac{A_{21}}{t_{23}^2} - \frac{A_{11}}{t_{13}^1} + \frac{t_{13}^1}{t_{23}^2} (A_{12} - A_{22}) \right]} \quad (31)$$

and

$$y_{23}^2 = \frac{1}{t_{23}^2} (\alpha_3 - t_{13}^1 y_{13}^1) \quad (32)$$

After x arrives at Region VI, it stays in Region VI if y_{11}^1 and y_{22}^2 are given by (C) and (D) and y_{13}^1 and y_{23}^2 are given by (31) and (32). Chattering is avoided.

5. CONCLUSIONS

This paper presents an extension to the earlier Kimemia and Gershwin work to add a real-time routing calculation to real-time scheduling. Thus this model can be used for many more types of manufacturing systems.

Future work will include the development of local operational rules which follow the system routing decisions calculated here, and extensive simulation of various types of industries to further demonstrate the use of this work.

REFERENCES

- R. Akella, Y. F. Choong, and S. B. Gershwin (1984), "Performance of Hierarchical Production Scheduling Policy," *IEEE Transactions on Components, Hybrids, and Manufacturing Technology*, Vol. CHMT-7, No. 3, September, 1984.
- R. Akella and Kumar (1986), "Optimal Control of Production Rate in a Failure Prone Manufacturing System," *IEEE Transaction on Automatic Control*, Vol AC-31, No. 2, pp. 116-126, February, 1986.
- R. Akella, O. Z. Maimon, and S. B. Gershwin (1987), "Analytic Approximation of the Upper Level Real Time FMS Scheduling", in preparation.
- D. Bertsekas (1976), *Dynamic Programming and Stochastic Control*, Academic Press, New York.
- S. B. Gershwin, R. Akella, and Y. F. Choong (1985), "Short-Term Production Scheduling of an Automated Manufacturing Facility," *IBM Journal of Research and Development*, Vol. 29, No. 4, pp 392-400, July, 1985.
- E. L. Hahne (1981), "Dynamic Routing in an Unreliable Manufacturing Network with Limited Storage," MIT Laboratory for Information and Decision Systems Report LIDS-TH-1063, February, 1981.
- J. Kimemia and S. B. Gershwin (1983), "An Algorithm for the Computer Control of a Flexible Manufacturing System," *IIE Transactions* Vol. 15, No. 4, pp 353-362, December, 1983.
- O. Z. Maimon and Y. F. Choong (1985), "Dynamic Routing in Reentrant Flexible Manufacturing System," MIT Laboratory for Information and Decision Systems Report LIDS-R-1554.
- Seidmann and Schweitzer (1984), "Part Selection Policy for a Flexible Manufacturing Cell Feeding Several Production Lines," *IIE Transactions* Vol. 16, No. 4, pp 355-362, December, 1984.
- J. N. Tsitsiklis (1981), "Optimal Dynamic Routing in an Unreliable Manufacturing System," MIT Laboratory for Information and Decision Systems Report LIDS-TH-1069, February, 1981.
- W. Whitt (1986), "Deciding Which Queue to Join: Some Counter-Examples." *Operations Research* v. 34, n.1, 1986, pp. 55-62.

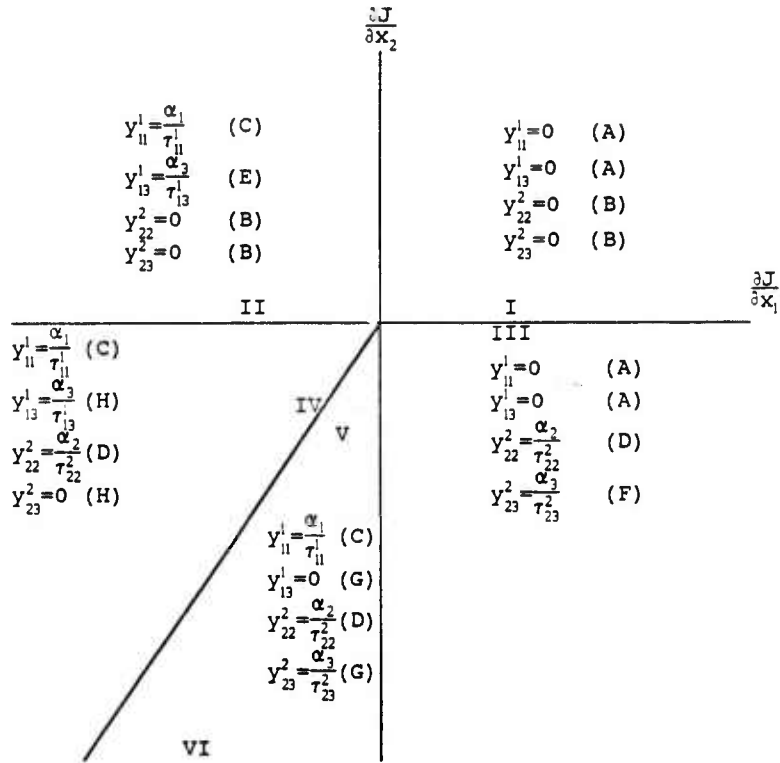


Figure 1. Control regions in $\frac{\partial J}{\partial x}$ space.

Computer-Aided Fabrication of Integrated Circuits

Paul Penfield, Jr.

**Professor of Electrical Engineering
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139**

**(617) 253-2506
penfield @ caf.mit.edu**

**Advanced Research in VLSI
Stanford University
March 23-25, 1987**

What is CAF?

Effective management of information associated with the fabrication of integrated circuits, in order to . . .

Improve:

**Flexibility
Portability
Quality
Yield
Drift**

Minimize:

**Turnaround time
Development cost
Confusion
Human errors
Manufacturing cost**



Who Benefits from CAF?

**Process Engineer
Machine Specialist
Facilities Manager**

**Operator
Manager
Circuit Designer**

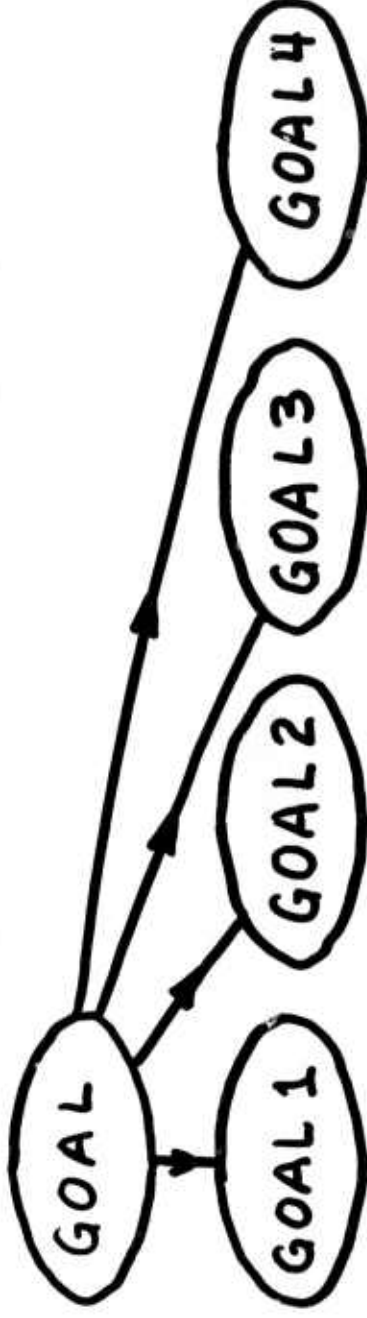
What Facilities Require CAF?

**Production
Foundry**

**Research Laboratory
Teaching Laboratory**

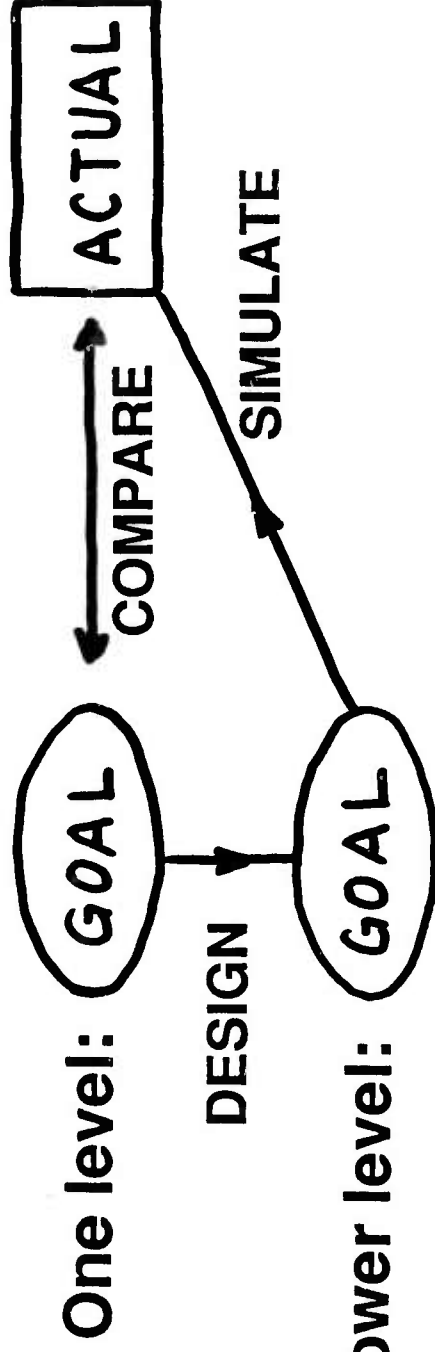
Two Design Strategies

1. DECOMPOSE (divide and conquer)



Add structure, same level. More objects, each simpler.

2. IMPLEMENT

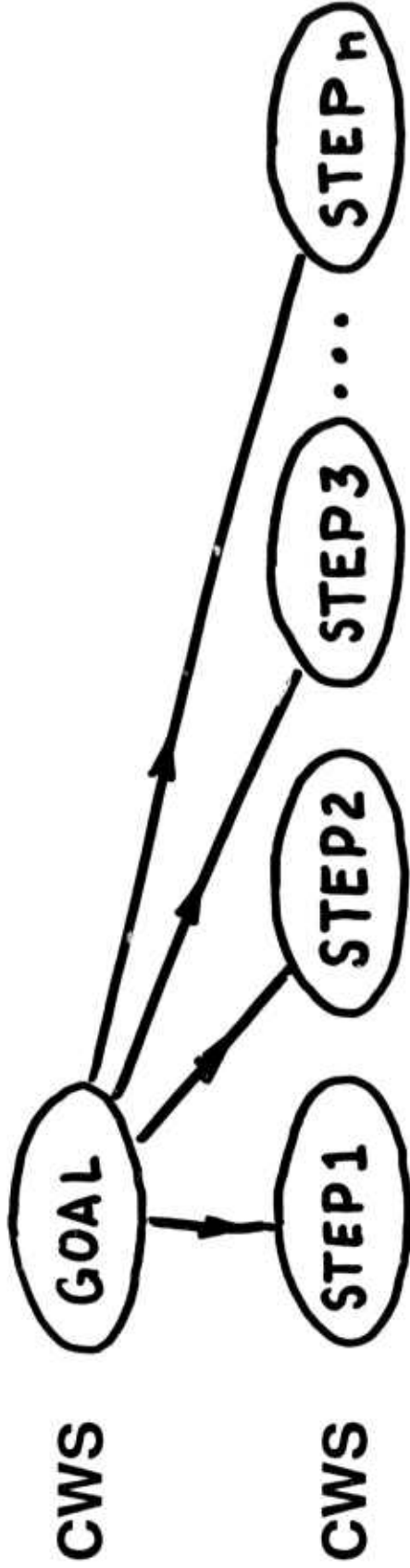


Express in lower-level objects, ultimately primitives.

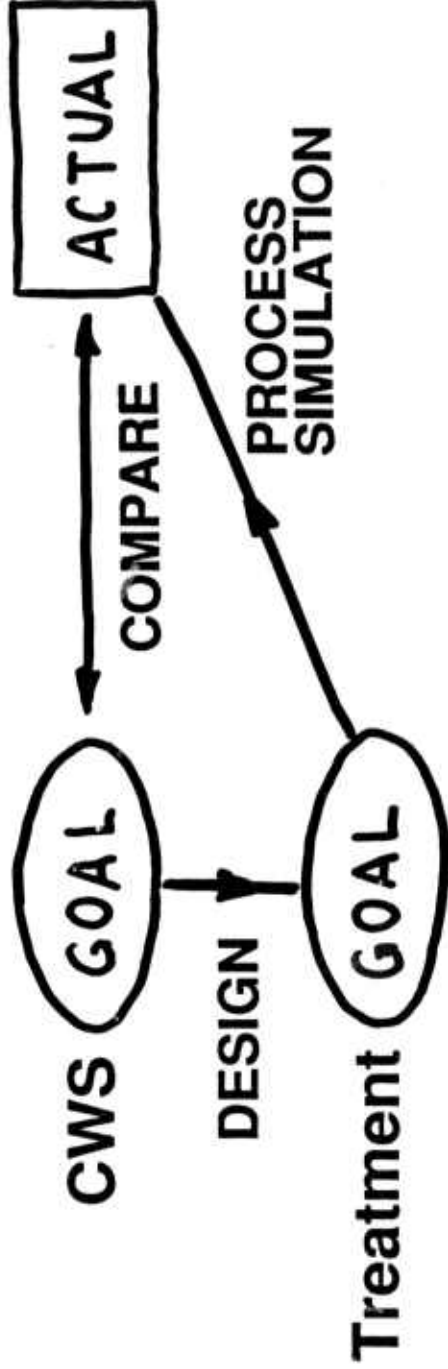
Semiconductor Process Design

Goal is Change in Wafer State

1. Decompose into successive process steps

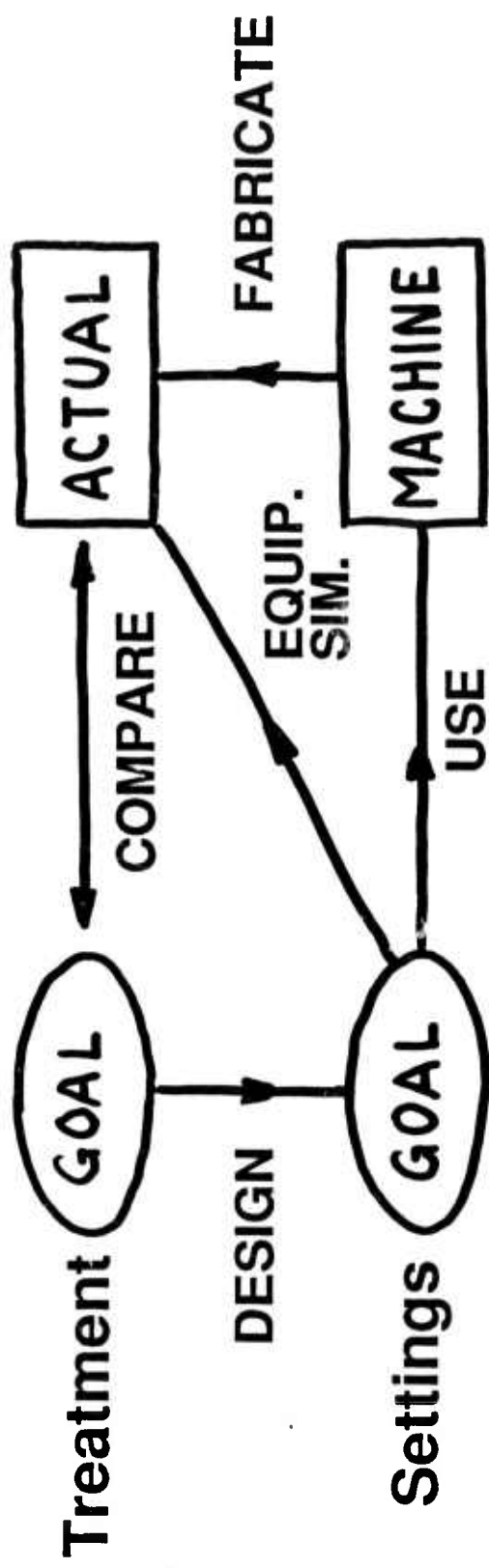


2. Implement each process step as wafer treatment



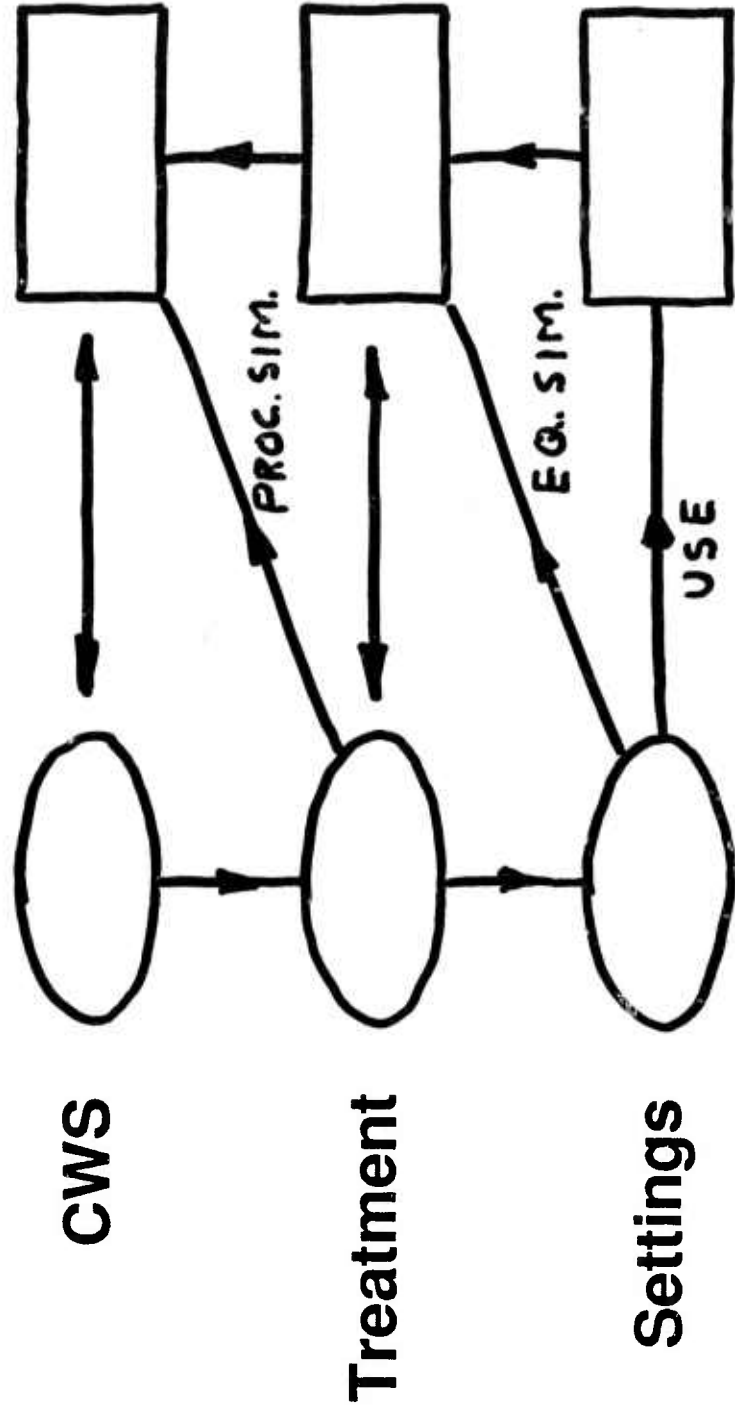
Semiconductor Process Design (Con't)

Implement each wafer treatment by machine settings, instructions or recipe

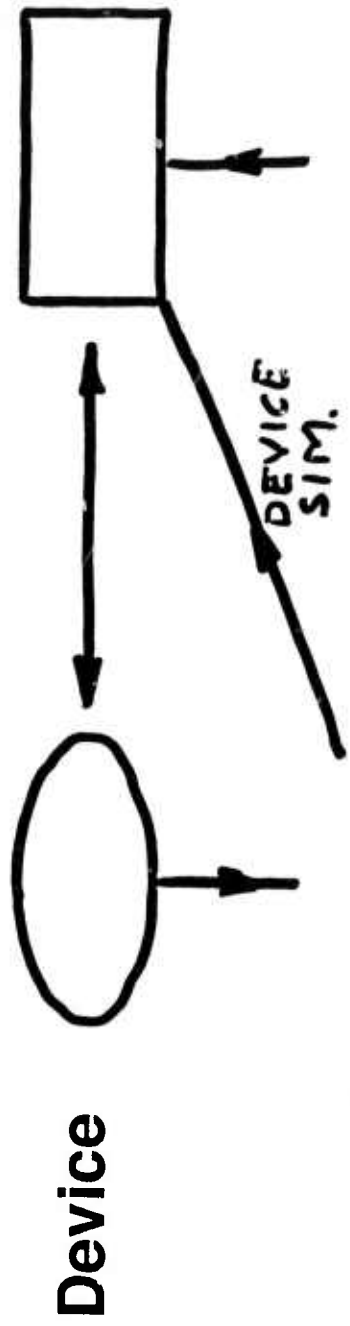


Settings are primitives for us since we are not designing the machines.

Two-Stage Process-Step Model



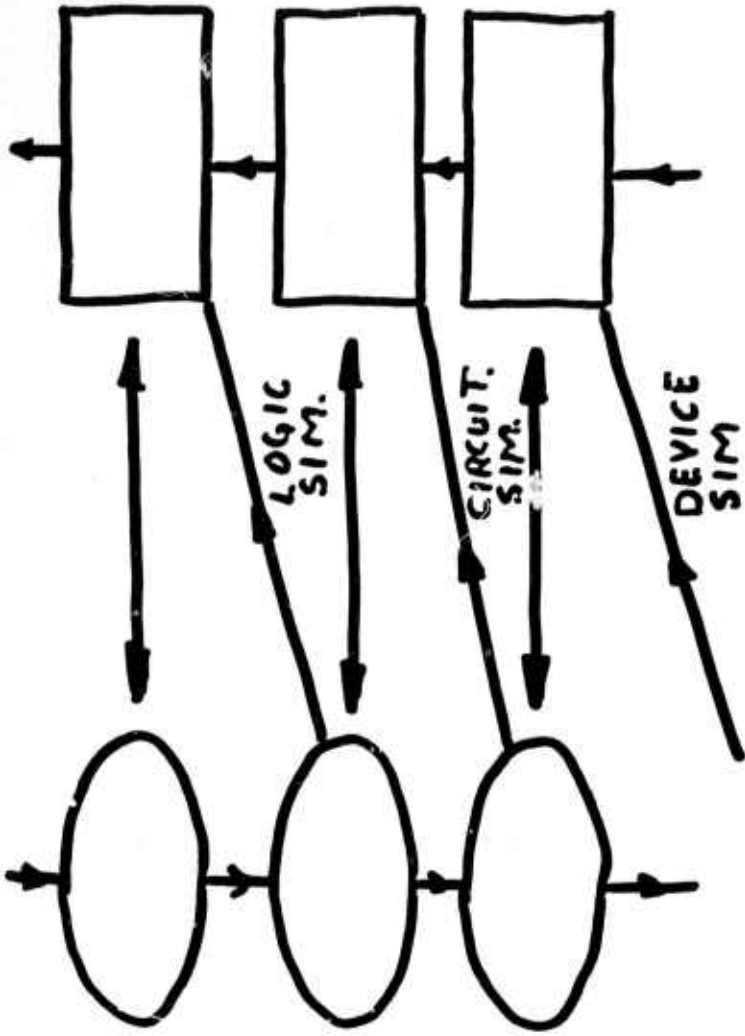
For the Process Designer



Logic

Circuit

Device



Facility Management (usual CIM)

Managers also do designs. What they design are schedules. Same design model applies. Levels are time horizons.

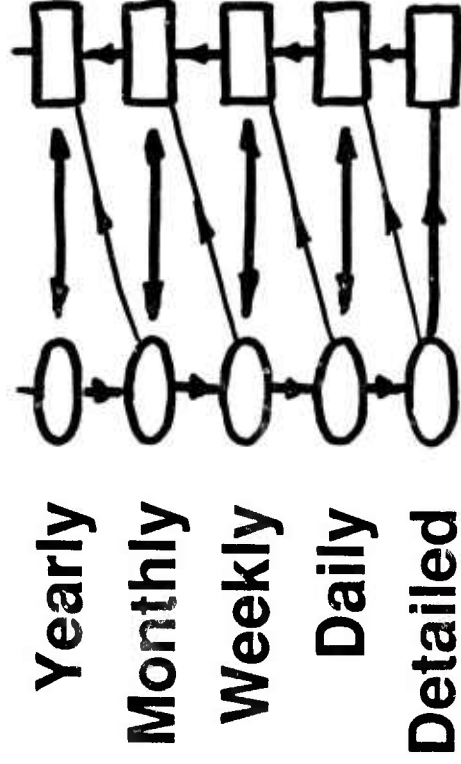
Production simulation predicts congestion.

Schedules are machine reservations, events.

Primitive level is detailed; others aggregates.

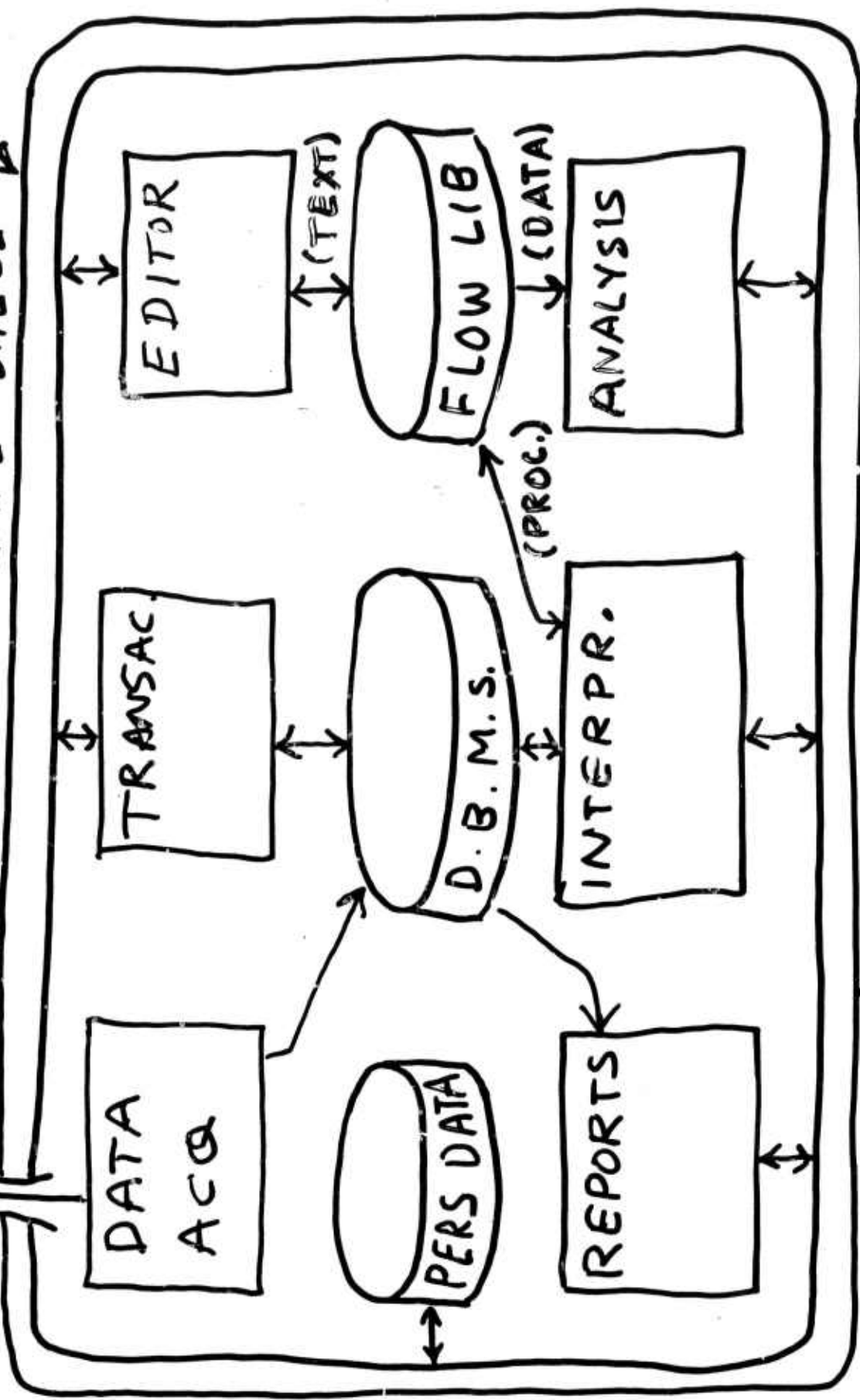
Issues: Machine failure/repair. Lot priorities.

Maintenance. Cost minimization.



MACH → SENSORS

CAFE SHELL



CAFE SYSTEM ARCHITECTURE



USER

Flow Language

3 levels: CWS, Treatment, Settings

Forms: Text, Data, Procedure

Drive several interpreters

Procedural abstraction. Minimal control structures

Splits and joins

Inspect/measure. Results accompany wafer objects

Multiple machines with separate settings

Mix scientific, production ideas

Wafer quantities. Control wafers

Deal with both CWS and WS

Under development. Related work at Berkeley

Data Management

New data model needed

Several new types

Intervals for variance control

Object-oriented access methods

Scientific and production data

Flows included

Access independent of storage form

Open architecture permits new reports

Model under development

Schema under development

Implemented in files and INGRES

Implementation Status

- Blue in use
 - Green in test
 - Red in dev.
 - Brown planned
 - Orange dream
-



Transactions

- Define facility
- Define machine
- Machine status
- Reservations
- Start-lot
- Authorized Users

...

Flows

- Baseline CMOS
- nMOS
- Integrated motor
- Undergrad lab process

...

Flow Editors

- **EMACS**

- **Template**

Flow Analysis

- **Minimum turn-around**

- **Laboratory policy adherence**

- **Is it safe?**

...



Interpreters

- **Walk-through**
- **Fabricator**
- **Production simulator**
- **Furnace model (equipment simulator)**
- **Process simulator link**
- **Device simulator link**
- ...

Personal Data

- **Notebook**

Reports

- **Alarm status, history**
- **Machine status**
- **Track-lots**
- ...



Data Acquisition

Machines:

- Dektak
- Ellipsometer
- Furnace tubes
- CV
- Clean station

...

Environment:

- Building alarms
- Door access
- Particle counter
- HVAC
- DI resistivity

...

User Interface

- **No paper in facility**
- **Roving shell (from Berkeley)**
- **Menu (from Berkeley)**
- **Forms-based interaction**
- **Automatic log in/out**

Summary

University renaissance in manufacturing.

Major research opportunities in CAF.

Integrate scientific, production ideas.

**Better documentation can help facility
operation and efficiency in many ways.**

Foundry can run custom process.

So for system designers . . .

**Custom parts, with custom processes,
by mail.**