Structured Design Methods for MEMS

Final Report

A Workshop Sponsored by
the National Science Foundation

Directorate for Computer and Information Science and Engineering (CISE)
Division of Microelectronic Information Processing Systems (MIPS)

November 12-15, 1995
California Institute of Technology
Pasadena, CA

Erik K. Antonsson, editor[*]
Engineering Design Research Laboratory
Division of Engineering and Applied Science
California Institute of Technology

October 15, 1996

[*] Associate Professor of Mechanical Engineering, Mail Code 104-44, Caltech, Pasadena, CA 91125, voice: 818/395-3790, FAX: 818/568/2719, e-mail: erik@design.caltech.edu

Attendees

Narayan Aluru	Massachusetts Institute of Technology
G. K. Ananthasuresh	Massachusetts Institute of Technology
Erik Antonsson	California Institute of Technology
Joseph Cavallaro	Rice University
Selden Crary	University of Michigan
Gary Fedder	Carnegie Mellon University
Michael Horton	University of California, Berkeley
Ted Hubbard	Technical University of Nova Scotia
Andrew Khang	University of California, Los Angeles
Frank Li	California Institute of Technology
Mark Long	California Institute of Technology
Paul Losleben	Stanford University
Ramaswamy Mahadevan	MCNC
Mary Ann Maher	Tanner Research
Richard D. Martin	Jet Propulsion Laboratory
Fariborz Maseeh	IntelliSense
Carlos Mastrangelo	University of Michigan
Linda Miller	Jet Propulsion Laboratory
Amar Mukherjee	University of Central Florida
Peter Parrish	Tanner Research
Kristofer Pister	University of California, Los Angeles
John Tanner	Tanner Research
Vance Tyree	Information Sciences Institute, USC
Peter Will	Information Sciences Institute, USC
Bernard Chern	National Science Foundation
Jack Hilibrand	National Science Foundation
John Staudhammer	National Science Foundation

The latest version of this report is available in PostScript format on the World Wide Web at:

`http://design.caltech.edu/NSF_MEMS_Workshop/`

The opinions, findings, conclusions and recommendations expressed in this Report are those of the authors and do not necessarily reflect the view of the National Science Foundation. The U.S. government has certain rights in this material. However, the material is not copyrighted and any part of this report may be reproduced, in any form and by any means, for dissemination of the ideas, information and recommendations contained in this Report.

Executive Summary

Over the last three decades, the microelectronics industry has undergone unprecedented growth and has had an enormous impact on the nation, in what is often called the "VLSI revolution". Research activities in both industry and academia have led to the rapid introduction of advanced semiconductor process technology, hierarchically structured design methodologies, automated design tools, simulation models and rapid prototyping techniques. One key to the rapid success of the VLSI development effort was the early definition (about 1970) of a clean digital interface that separated design efforts at increasingly high levels of abstraction from the growing complexities of the fabrication processes. This allowed the designer to focus on process-independent design tools and methodologies that are available to research and academic community for rapid prototyping of VLSI chips.

Though there have been many remarkable and revolutionary advances made in Micro--Electro/Mechanical Systems (MEMS) design and fabrication during the past decade, the need for structured design methods remains. At present each new MEMS development is expensive ($1M or more), and time consuming. One contributing factor is that there is not yet an equivalent to the Caltech Intermediate Form (CIF) or the other descriptive languages which are commonly used in VLSI design. MEMS fabrication processes have matured rapidly but they are still many and varied. The time is now ripe to define and develop structured design methods, and to take advantage of the still formative nature of much of the field. The experience of the VLSI research community of 25 years ago in evolving design methodologies and fabrication processes should provide useful guidance.

Several elements appear to contribute to the successes in developing structured design methods for VLSI:

- a small (but growing) number of functional elements,
- a largely planar topology,
- a largely rectangular (Manhattan) geometry,
- the independence of form and function,
- conservative design rules can eliminate complicating effects,
- a description of function exists (Boolean logic),
- "There is a clean separation between the processing done during wafer fabrication and the design effort that creates the pattern to be implemented." [C. Mead and L. Conway, *Introduction to VLSI Systems* Addison-Wesley Publishing Company, 1980, Page 47.]

The design of mechanical systems, more generally, appears to have none of these virtues, which largely explains the lack of structured design methods for mechanical design. However, MEMS is a hybrid of VLSI and mechanical systems, employing the materials and fabrication processes of the former, while utilizing many of the energy storage and transfer domains of the latter. This report concludes that while significant differences between digital VLSI design and MEMS design clearly exist, sufficient parallels also exist to strongly encourage research on structured design methods for MEMS.

Three areas currently in use in VLSI design have also been identified as common elements for structured MEMS design:

Languages for interchange of data among designers and between designers and fabricators,

Libraries for storing previously successful MEMS device designs for reuse, and

Simulation of desired function and of the fabrication processes.

Each of these has played a crucial role in the development of design methodologies for VLSI, and building on those prior VLSI developments will form the basis for a “clean separation” between design and fabrication of MEMS, and will provide the greatest leverage from an investment in research resources. Advances in these three areas will provide the foundation for (semi-)automatic synthesis of MEMS, perhaps by compiling a schematic or language describing desired function into a set of masks and processing information to fabricate a device or system that will robustly exhibit the desired function.

While the challenges in developing structured design methods for MEMS that preserve the “clean separation” are significant, the benefits of such methods will greatly enhance advances in MEMS. Developing structured design methods for MEMS holds the promise to significantly reduce the costs and time to create new devices and systems, and increase the complexity and robustness of devices and systems that can be designed.

Contents

List of Figures

List of Tables

Preface

This workshop arose from discussions at the preceding two NSF sponsored workshops: *New Paradigms for Manufacturing* held at the National Science Foundation on May 2-4, 1994 [1], and *Design Methodologies for Solid Freeform Fabrication* held at the Engineering Design Research Center at Carnegie Mellon University on June 5-6, 1995 [2]. These workshops examined the idea of applying VLSI-like design methods to selected areas of mechanical design. These discussions suggested that the two areas to explore first are SFF and MEMS.

The workshop that produced this report: *Structured Design Methods for MEMS* held at the California Institute of Technology on November 12-15, 1995, builds on the earlier: *Small Machines, Large Opportunities: A Report on the Emerging Field of Microdynamics*, Report of the NSF Workshop on Microelectromechanical Systems Research, 1988 [3], and explored a research agenda for extending VLSI-like design methods to MEMS.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. MIP-9529977. Any opinions, findings, conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the sponsor.

The encouragement and support of Bernie Chern, Amar Mukherjee, John Staudhammer, and Jack Hilibrand is gratefully acknowledged. Many thanks to Kris Pister for his many invaluable contributions to the organization of this workshop. Kris Pister, Joe Cavallaro, Gary Fedder, Amar Mukherjee, and Jack Hilibrand were all pressed into final review and editing tasks.

We are all indebted to the workshop participants for the contribution of their time and thoughts to the discussions, and to this Report.

Arrangements and details were magnificently arranged by Charmaine Boyd.

1 Background of the Workshop

1.1 Introduction

On November 12-15, 1995 a workshop sponsored by the National Science Foundation was held at the California Institute of Technology to discuss and explore the research issues involved in developing structured design methodologies for Micro-Electro-Mechanical Systems (MEMS). The Workshop gathered attendees from many of the major research universities involved in MEMS fabrication and MEMS design research, as well as several industry representatives. The goal of the Workshop was to identify areas of profitable fundamental research in the area of structured design methods for MEMS. The primary question posed to the Workshop was the following: Can the successes in developing structured design methods for digital VLSI be extended into the domain of MEMS? If so, what lessons can be learned and transferred?

The workshop attendees were divided into four discussion groups, and a series of questions were suggested to each group:

- The **Synthesis** group's discussion encompassed the development of methods for automatic or semi-automatic generation of MEMS shapes and/or masks.
  - What type of design rules can be defined that when applied to the representation will guarantee successful fabrication in a series of MEMS fabrication processes?
  - What shape synthesis tools are needed?
    - Can methods be developed to automatically create masks (and other fabrication information) to fabricate a desired shape?
  - What function synthesis tools are needed?
    - VLSI design has benefited by building complex functions from hierarchies of simpler elements, including automatic mapping of logical constructs into physical devices and physical placement of devices and wire routing
    - Can the same approach make progress towards automatic determination of shape(s) that will exhibit a desired function?
  - Can a methodology be developed to compile a schematic representation into masks and processing information to fabricate a device or system?
- The **Function Simulation** group considered approaches and requirements for simulation of the mechanical and electrical function of the resulting devices and systems.
  - What description(s) of function should be developed?
    - Should libraries of functional elements be compiled?
    - Should a small set of functional sub-elements be developed, from which more complex systems can be built?

  - What function simulation tools are needed?
    - Considerable work has been done on FEA of stress, strain, and electrical charge for MEMS.
    - What other simulation of function will be important?

- The **Digital Data Interchange Languages** group discussed the requirements for languages and standards for interchange of information between research, design, fabrication, and testing groups.
  - What level or levels of abstraction should be used for describing the physical design?
    - Should it be two-dimensional layers corresponding to the masks which are used during the fabrication process?
    - Or should it be a form of a three-dimensional description?
    - Or should it be a descriptive hierarchy with tools to move between the levels?
  - What type of model should be used to represent MEMS designs?
    - What should be the role of traditional solid modeling?
    - What attributes should the model provide in describing the design (in addition to geometry)? Potential candidates include strength, material, microstructure, tolerance, etc.
  - What should be the form of a digital design exchange format?
    - Can the format support alternative MEMS processes?
    - Is there a common set of information required by all MEMS processes?
    - What formats and design methodologies are in use today to support the MEMS processes and is there a generic methodology applicable to all (many of) the MEMS processes?
    - Is there some grouping of MEMS processes which will facilitate the identification of generic methodologies?

- The **Fabrication Process Simulation** group examined the requirements for the simulation of MEMS fabrication processes during the design process.
  - What fabrication simulation tools are needed?
    - The physical device is not a simple extrusion of the 2-D mask
    - Is there a need for fabrication process planning (or can a “clean separation” between the design activity and the fabrication details be maintained)?
    - Are design critics to identify non-manufacturable features a valuable approach?

These four groups met over the course of the four day workshop, with twice-daily meetings of the whole group for discussion and interchange. The reports of the four discussion groups are included in Section 3 of this report. The whole-group discussions helped refine the group discussions, developed some common themes, and also generated an overarching agreement on the necessity for some infrastructure development. A brief report on suggested infrastructure developments in included in Section 3.

A brief introduction to the background of this report is included in the next section, followed by brief summaries of each discussion group's findings and recommendations, followed by the full report of each group.

Each participant was encouraged to write a position paper in conjunction with the workshop. These are attached to this report in Section 4.

1.2 Structured Design Methods

1.2.1 Structured Design Methods for Digital VLSI

Beginning in the late 1970's, representatives from universities and industry convened a series of workshops on Very Large Scale Integrated Circuits (VLSI) in order to bridge the gap between abstract design and physical fabrication. Until that time, translation from the abstract logic circuit to a manufacturable layout was a labor-intensive process. Designing the physical layout required expertise in the fabrication process. Due to the laborious translation, logic designers received little feedback on the cost and performance consequences of their design decisions.

The series of VLSI workshops promoted the use of a fabrication-independent, geometric representation called CIF (Caltech Intermediate Form). Designers could produce CIF representations of their logic designs and apply simple geometrically-oriented design rules. If the CIF description met all the geometrical design rules, the design was guaranteed to be manufacturable. A number of silicon foundries would accept CIF inputs and produce fabricated chips. The CIF description would be translated automatically into detailed fabrication instructions meeting all constraints of the foundry's fabrication process.

Several benefits derived from the development of CIF and the associated design methodology it enabled:

- A standard interchange format between designers and fabricators enabled numerous Computer-Aided Design (CAD) tools to be generated and distributed. These tools helped fuel the commercial CAD tool industry.

- Universities could teach process-independent VLSI design to graduate and undergraduate students, providing students with feedback on the results of their design decisions.

- Foundry services were developed which could translate CIF files into working silicon.

- Both students and industrial chip designers could use the foundries to produce working integrated circuits on "the first silicon".

1.2.2 Structured Design Methods for Mechanical Systems

An NSF-sponsored workshop was held on May 2-4, 1994 entitled *New Paradigms for Manufacturing* [1] to determine if it was feasible to define an equivalent design methodology for mechanical systems (or some subset of mechanical systems), decoupling the design representation from the fabrication process. The participants in the workshop concluded that there were enough potential similarities between the VLSI design and some classes of mechanical design for rapid prototyping to warrant further investigation. In particular, two new classes of layered manufacturing processes were identified as having strong similarities with VLSI fabrication: micro-electronic mechanical systems (MEMS) and solid freeform fabrication (SFF). MEMS employs many of the same manufacturing steps as VLSI and the MEMS community seems especially amenable to examining the VLSI design methodology to adapt it to their tasks. This workshop explored the potential for a unified design methodology aimed at supporting the MEMS fabrication technologies.[1]

1.2.3 Structured Design Methods for MEMS

Motivated by exciting early work in the area [6], in 1988 the National Science Foundation sponsored a series of three workshops on Microelectromechanical Systems Research. These workshops resulted in a report entitled: *Small Machines, Large Opportunities: A Report on the Emerging Field of Microdynamics* [3]. This report initiated NSF funded research in the MEMS field. The motivation described in the report remains viable today.

> “The miniaturization of electronics has produced a far-reaching technological revolution. Now mechanics is poised on the brink of a similar miniaturization, and its own revolution. Researchers are working toward creating microdynamical systems, the microscale derivatives of conventional large-scale electromechanical systems.” [3, page 1]
>
> “The technology of microdynamics is based on that of microelectronics but calls for important advances over it. The goal is to make fully assembled devices and systems that can do what large-scale electromechanical systems cannot do as cheaply, or at all.” [3, page 1]
>
> “In recent years these techniques have provided the basis for a viable and growing sensor industry. This industry’s greatest commercial success is pressure sensors for automobiles.” [3, page 4]
>
> “Current mask design and creation programs were written in response to the fabrication requirements of silicon-based electronic devices and are now highly optimized for the technology of microelectronics. Many of the program features are, at best, useless and, at worst, contrary to the needs of silicon mechanical device fabrication.” “Current programs typically [create] rectangular features arranged in Manhattan grids.” “. . . pattern features other than rectangular, for example, curvilinear and freeform, will be necessary for making microdynamical items such as springs, gears, and bearings.” [3, page 15]
>
> “Microfabrication technologies, based on batch fabrication, lithography, and selective etching, impose new constraints on the design process. The conventional iterative fabrication [and] sequential refinement ... [is] inappropriate [for MEMS]. The entire design must be completed before fabrication is begun.” [3, page 16]
>
> “Thus, simulating designs before they are fabricated, as is done in electronics and in large-scale mechanics, is highly beneficial. A set of computer-aided

[1] A previous NSF sponsored workshop held on June 2-5, 1995 at Carnegie Mellon University investigated the issues of design methodologies for SFF. [2]

> design tools can reduce the overall cost and/or time between conception and prototype and improve designs for better performance. However, for several reasons, these simulation tools are not now readily used in microdynamical systems design.” [3, page 16]

Though there have been many remarkable and revolutionary advances made in the MEMS area since 1988, the need for structured design methods for MEMS remains [4]. For example, there is not yet an equivalent to CIF or the other descriptive languages which are commonly used in VLSI design. The MEMS fabrication processes are maturing rapidly, but they are many and varied. The time is now ripe to develop structured design methods, and to take advantage of the still formative nature of the field. The experiences of the VLSI research community of 25 years ago in developing design methodologies and fabrication processes should provide useful guidance.

Several elements appear to contribute to the successes in developing structured design methods for VLSI:

- a small (but growing) number of functional elements (resistors, capacitors, transistors, shift registers, logic gates, memory cells, data registers, adders and other arithmetic logic units, processor core elements, etc.)
- a largely planar topology (Manhattan geometry but with up to three layers of interconnection of elements)
- a separation of form and function (*e.g.*, the function of a device is independent of its position and orientation on the wafer)
- conservative design rules can eliminate complicating effects (for example, only scalar flows of electrons are considered in digital VLSI design, concerns for the 3-D vector and tensor quantities are eliminated)
- a description of function exists (in VLSI’s case: digital electronic function)
- “There is a clean separation between the processing done during wafer fabrication and the design effort that creates the pattern to be implemented.” [5, page 47]

Many have long argued that macroscopic mechanical design has none of these virtues, which largely explains the lack of structured design methods for mechanical design [1]. The question remains, however, is MEMS enough like VLSI so that structured design methods can be developed? If so, which part or parts of the developments in VLSI will translate to MEMS?

One approach to developing these new design formalisms and methods is shown in Figure 1. The boxes represent physical artifacts, the arrows represent processes. The central straight line indicates: a mask is processed (by fabrication) to create a shape, which when put into operation exhibits a function. The lower arrows represent engineering analysis. For example, a simulation of the fabrication processes can be used to process a mask-layout

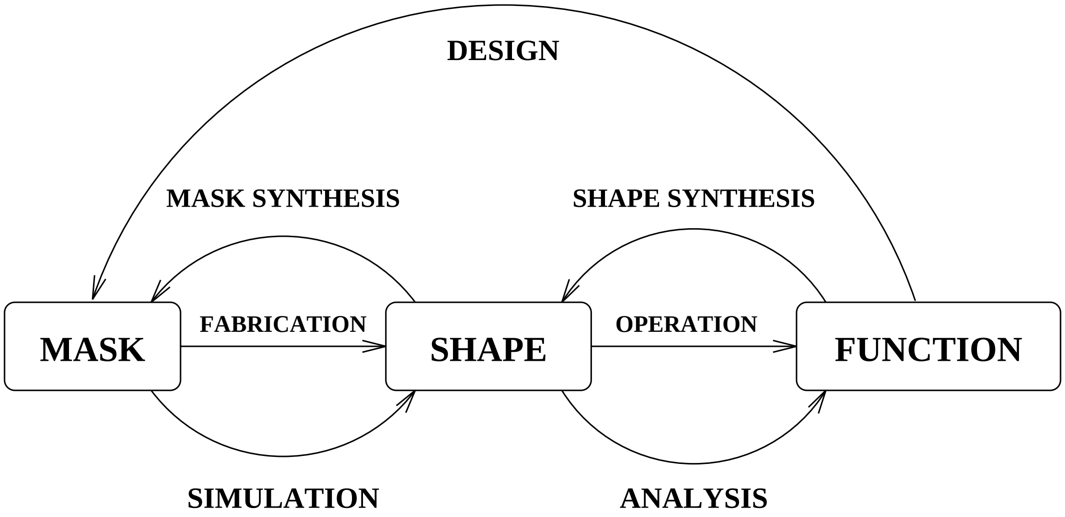
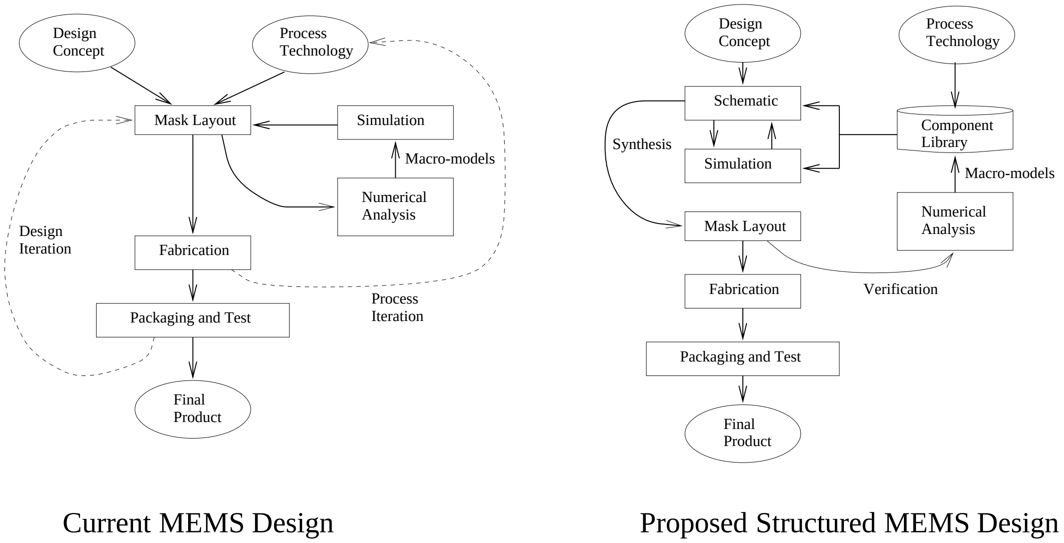


Figure 1: Design and Analysis Processes.

into a geometric model of the expected shape, which can be analyzed with a finite element method to predict the function of the device. The “backwards” pointing arrows in the upper portion of the figure represent synthetic (rather than analytic) processes, corresponding to creating a shape that will exhibit a desired function, and generating a mask that will create a desired shape.

An alternative view of this general approach is shown in Figure 2. The left-hand flowchart in Figure 2 shows a typical MEMS design process of today. The right-hand flowchart combines a number of new structured approaches, including reusable component libraries, advanced simulation and analysis, and (similar to Figure 1) a direct synthesis of layout from schematic.

While the challenges to developing structured design methods for MEMS are significant, the benefits of such methods will greatly enhance MEMS developments. A set of initial research steps appear to be reasonably clear, and are discussed in the following sections of this report.



Current MEMS Design

Proposed Structured MEMS Design

Figure 2: MEMS Design Flowcharts.

1.3 Charge to the Workshop on Structured Design Methods for MEMS

Dr. Bernard Chern
Division Director
Microelectronic Information Processing Systems (MIPS)
National Science Foundation
4201 Wilson Blvd.
Arlington, VA 22230
tel: 703/306-1940, Fax: 703/306-0610
bchern@nsf.gov

The goal of research in rapid prototyping is to develop and integrate methodologies, tools, environments and technologies needed to be able to automate the rapid, efficient and accurate design and construction of processes, artifacts and systems of artifacts. A key long term objective is to develop a design methodology that can be applied generally to mechanical and electromechanical systems.

This workshop is intended to identify and encourage research efforts on implementing MEMS design methodologies. The scope of these efforts, however, includes not just the design methodology itself, but also the design tools, design environments and design technologies that will be available for rapid prototyping using MEMS.

At this workshop we will grapple with research needed to create a VLSI-like design methodology (including tools, environments, levels of abstraction, . . .) for the MEMS technologies in which there is a clean separation between design and fabrication; that is, between the design community with its concerns about CAD tools, design environments, etc., and the fabrication community with its concerns about equipment and processing capabilities as well as such customer servicing criteria as cost and responsiveness.

To achieve such an outcome we will need to answer some key questions:

- Can we use the same generic layering model for all MEMS fabrication processes? Should it differ from “the most generic model” described in C.A. Mead’s Preface to the Workshop on *New Paradigms for Manufacturing*?

- Can we identify a common digital specification language that can be used generally to describe the desired prototype in terms of the resulting geometry on each layer? If the answer is CIF (Caltech Intermediate Format), is CIF adequate for present and future systems?

- Can we use this digital interface (consisting of CIF, the generic model of layered fabrication and the common understanding of the standard process) to achieve a clean separation between the design and processing activities? (See R. Sproull’s paper: “Digital Interfaces to Fabrication” from the Workshop on *New Paradigms for Manufacturing*.)

- What benefits could be derived from the use of 3D modeling in the MEMS design methodology? In SFF, design is done using 3D modeling (SIF - Solid Interchange

Format). The 3D model is then sliced yielding data in L-SIF form (L-SIF - Layered SIF). Is the final physical shape part of the MEMS design process or is the design carried out in terms of function and 2D descriptions in CIF? How does the MEMS design methodology relate to that for SFF? (See paper by C.H. Sequin and S. McMains titled "What can we Learn from the VLSI Revolution?" from the Workshop on *SFF Design Methodologies*.)

- Can we make the design tool hierarchy independent of process evolution (smaller feature sizes, thinner polysilicon layers, etc.)? In the VLSI domain the fabricators are driven by the design community through interaction across the "clean separation" interface; can this be done in MEMS?
- What levels of abstraction are appropriate to MEMS mechanical design and how can we improve the ease of moving among these levels?
- Does the current design methodology used in MEMS enable a compatible treatment of the electronic and mechanical structures in the MEMS system? Would such a treatment be advantageous?
- Does today's MEMS design methodology start with function or with shape? SFF design starts with shape (3D model) but VLSI starts with function (HDL description). Which approach is best, in MEMS applications, for the designer, the fabricator, the user?
- How much commonality is possible between MEMS design and SFF design: language? interface? levels of abstraction? design tools? etc.
- The MCNC/MUMPS infrastructure has been successful in establishing a standard process and making it widely available. What can be done through improvement of the MEMS design methodology to enhance the potential for rapid, error-free product generation within the present process capability?

A successful workshop for developing a common design methodology for the MEMS technologies will result in:

- A strong argument for an NSF/MIPS research program focused on systems design to make for the rapid prototyping of MEMS structures.
- Definition of a common low-level layer-based digital interface descriptive language for present and future MEMS technology implementations. This can be a great advantage, making a variety of MEMS technologies available to the designer (without learning a new interface language) and making many customers available to the MEMS fabricator without the need to invest time and energy bringing the designer up to speed to address his technology. Further, modifications and advances in the MEMS process area could be accommodated within the same design framework, merely involving changes in the design rules. Also the digital interface language would permit

design submission over a network so that brokerage services, such as MOSIS and MCNC, become practical.

- Steady accumulation in the industry of feature and object description libraries, in a common language, that can be incorporated in the design heritage of the field and that will encourage the refinement of a hierarchical design methodology that will make them useful to the entire design community.
- Momentum to create software paths from the higher level descriptions (with which the designer starts) to this digital interface language.
- Momentum to create paths (algorithms and translation code) to go from this common digital interface language to the languages that drive the MEMS fabrication systems.
- Refocusing by those active in MEMS-based prototyping from the nuts and bolts of the design process to the optimization of system implementation along the dimensions of time to market, multiple sourcing, reuse of design building blocks, de-skilling of low level design activities, etc.
- Lower cost, reduced delay and fewer errors for system implementation using MEMS-based rapid prototyping technology. (Lower cost because the design is cheaper and more designs go down this design path to many competitive vendors, reduced delay and errors because the design path is significantly automated and takes advantage of design heritage)
- Increased exploration of sophisticated MEMS product design alternatives because the time and cost for experimental implementation are brought within reach so that there will be more cycles of learning about market requirements incorporated in each product generation
- These techniques for using a low level digital descriptive interface in MEMS can (potentially) and will (very likely) be translated into enhancements for more general mechanical system design (*e.g.,* SFF) where they will provide an impetus for extending design automation to ever-widening regions of mechanical design space

1.4 Discussion Groups

The workshop attendees, identified by discussion group, are listed below.

- **Synthesis** Discussion Group Members:

  Joseph Cavallaro (Rice University), Moderator
  G. K. Ananthasuresh (MIT)
  Michael Horton (Berkeley)
  Andrew Khang (UCLA)
  Mark Long (California Institute of Technology)
  Carlos Mastrangelo (University of Michigan)
  Peter Parrish (Tanner Research)
  Kristofer Pister (UCLA)

- **Function Process Simulation** Discussion Group Members:

  Selden Crary (University of Michigan), Moderator
  Narayan Aluru (MIT)
  Gary Fedder (Carnegie Mellon University)
  Ramaswamy Mahadevan (MCNC)
  Mary Ann Maher (Tanner Research)
  Richard D. Martin (Jet Propulsion Laboratory)
  Linda Miller (Jet Propulsion Laboratory)
  Peter Will (ISI)
  John Staudhammer (NSF)

- **Digital Data Interchange Languages** Discussion Group Members:

  Vance Tyree (ISI), Moderator
  Erik Antonsson (California Institute of Technology)
  Paul Losleben (Stanford University)
  Amar Mukherjee (University of Central Florida)
  Bernard Chern (NSF)

- **Process Simulation** Discussion Group Members:

  Ted Hubbard (Technical University of Nova Scotia), Moderator
  Erik Antonsson (California Institute of Technology)
  Frank Li (California Institute of Technology)
  Ramaswamy Mahadevan (MCNC)
  Fariborz Maseeh (IntelliSense)
  John Tanner (Tanner Research)
  Jack Hilibrand (NSF)

2 Summary Findings and Recommendations

This section presents a summary of the findings and recommendations of each of the workshop discussion groups. While each of the groups addressed a different topic, the findings and recommendations contained some striking commonalities. Nearly every group included recommendations to encourage and support research into:

- data interchange representations/**languages**,
- **libraries** of successful MEMS (sub-)elements,
- analysis and **simulation** in multiple energy domains.

Thus it appears that these three areas may form the core of structured design methods for MEMS.

However, the reports (perhaps properly, at this early stage of development) do not contain any specifics on the information that will be interchanged by the languages, stored in the libraries, or analyzed across multiple energy domains. Therefore, a top-level issue remains to be clarified, namely: what information should be used to represent the MEMS designs (either designs in progress, or completed devices). Should it be the attributes of the final desired device (*e.g.*, 3-D shape and materials), or should it be the layout of the masks and other processing information to create the desired final (3-D) shape? Should it be *function* or *structure*?

Developing Structured Design Methods for MEMS holds the promise to significantly reduce the costs and time to create new devices and systems, and increase the complexity and robustness of devices and systems that can be designed.

2.1 Synthesis Summary

Synthesis proceeds from a high level behavioral or structural description of the desired object or system to a detailed geometric description suitable for transmission to a fabricator. There has been great progress in VLSI synthesis based on elementary Boolean building blocks (simple gates) and higher level parameterized functional blocks (registers, core processors, etc.). Such progress has been of greatest value to system level designers who can design layouts at a high level of (functional) abstraction. The synthesized designs are not as efficient as simple designs implemented by experienced chip designers, but they are generally used for highly complex designs where the potential for errors, working at the lowest level of abstraction, is a severe problem. What began with simple parameterizable building blocks (like multistage shift registers) was steadily broadened until today full complex functions are used as building blocks in still more complex applications.

The richness of possible applications for MEMS and their growing complexity suggest that there is great potential for developing a synthesis technology. It will be necessary to start at the simplest possible level with a focus on descriptive languages which are not limiting but which are amenable to becoming the basis for a hierarchy of building blocks. One example might be a mechanical filter in which the multiple pass and stop bands can be synthesized of independently parameterized elements tied together. The gradual accumulation of a library of building blocks depends on maintaining a common descriptive language over a long period of time and making the descriptions in that language as generic as possible. One thrust of the synthesis effort, in the absence of generic building blocks (Boolean logic functions in digital VLSI), is to accumulate design heritage in libraries of building blocks with geometric, compositional, functional and performance data included.

Finding: Languages for representing MEMS functions do not yet exist. At present an HDL (Hardware Descriptive Language) for MEMS is not available. A language is required whose syntax will facilitate the mapping of function and other attributes to shape, perhaps via the building blocks in the MEMS cell library.

Recommendation: Support the development of languages for MEMS design, perhaps including a MEMS HDL. Encourage MEMS/HDL language standardization and use in research and education.

Finding: Libraries of reusable MEMS (sub-)elements, devices and systems are beginning to be formed.

Recommendation: Support the development of common descriptive languages and interfaces so that system designers can take advantage of existing custom designed component building blocks in their system design activities. Support the development of tools for formal verification and assembly using existing library elements.

Finding: In order to develop highly complex and integrated MEMS, synthesis methods are required that can start with a multi-energy domain schematic and create component shapes.

Recommendation: Support research in the definition of multi-domain representation and extended semantics for links between the domains; support the development of tools for shape generation, search, placement and routing of multiple objects, etc.

Recommendation: Support efforts to (1) broaden the class of devices with synthesis from performance description to configuration and (2) to identify other classes of functions that are amenable to this simplified treatment.

2.2 Function Simulation Summary

Simulation of the mechanical and electrical function of MEMS devices and systems is intended to provide the designer with the ability to insure that the design he submits for fabrication will meet the function and performance requirements of the application. The representation of the device occurs at many levels of abstraction and it is important to be able to move between levels accurately and unambiguously to support top-down and bottom-up design.

Equally, a CAD framework for MEMS needs to support representation and analysis in multiple energy domains (electrical, mechanical, thermal, radiant, chemical, magnetic, acoustic and fluidic) with coupling between them. In each of these domains, the MEMS system must be represented so that function can be verified at the desired level of abstraction. The design framework required for MEMS is complex, but it is important to define the framework properly so that design can be performed effectively and accurately by novice and expert designer.

Finding: A common language for representing MEMS functions is needed.

Recommendation: Support the development of a theoretical framework and tools for representation and analysis of (multi-energy domain) MEMS functions, and make the tools available to the design community. Some such tools, by analogy to VLSI, might be:

- Layout verification tools (design rule checkers, component extractors, layout vs. netlist comparers, etc.)
- Hierarchical schematic description and editing
- Multidomain simulators
- Circuit level simulators (Spice-like equations with parameter extraction capabilities)
- MEMS specific solid modelers capable of being manipulated to provide object descriptions suitable for transmission to the fabricators
- Hardware behavioral modeling language (similar to VHDL-A?)

Finding: Similar to VLSI, libraries of MEMS functional (sub-)elements appears likely to facilitate development of complex systems.

Recommendation: Establish a framework for libraries of previously successful MEMS devices and elements.

2.3 Digital Data Interchange Languages Summary

A widely used language for (digital) interchange of MEMS design data is crucial to rapid advances. The theme of “representation framework” or “language” is common to the five areas considered by this workshop, and underlies nearly all of the findings and recommendations. Synthesis, simulation of function, simulation of fabrication processes and teaching all will depend on an appropriate representation for interchange of information.

The goal of any such interchange representation is to permit the sharing of design knowledge and experience, as well as the rapid transmission of design information to fabricators. The interchange representation can also serve an additional function: that of separating the design process from the fabrication process. Should such a “clean separation” be achievable, it will arise through the construction and use of an appropriate information interchange language. Any such language must accommodate current MEMS (sub-)elements, devices and systems, while not constraining future developments.

One crucial aspect to resolve early is the type of information to be transmitted. In VLSI, designers transmit the desired final 2-D shapes for each layer to the fabricators. The fabricators then (with an intimate understanding of their processes) adjust or compensate or pre-distort the mask-layouts so that the desired final shapes will be created. Currently in MEMS, designers do the pre-distorting and must therefore have a deep understanding of the fabrication processes to be used. This results in much experimentation and many prototype fabrication cycles. The current strong connection between design and fabrication for MEMS is unlike digital VLSI, and may prove a hindrance to rapid advances.

Finding: An extensible standard digital data interchange format or language is needed for MEMS.

Recommendation: Since the interchange representation underlies nearly all other design-related activities for MEMS, it is crucial that research in this area be vigorously initiated and pursued. This effort should not be limited to shape alone, but should include materials properties, layer properties, function, etc., and provide data for all of the required analyses (stress/strain, electrical charge, fluid flow, vibration/mode-shape, etc.).

Finding: Improved descriptive language for 2-D objects is needed capable of describing non-polygonal geometries and interfacing to modern mask-layout generators.

Recommendation: Support the development and adoption of extensions to the CIF (or GSDII) language to accommodate non-polygonal geometries, to provide for a variety of geometry adjustments suitable for the MEMS processes (shrinks, bloats, etc.) and to interface to mask making equipments.

2.4 Fabrication Process Simulation Summary

Process simulation is the conversion of multiple two-dimensional layout geometries and process information into a three dimensional representation of the output shape. It is used to provide feedback in the design/fabrication loop before the full commitment to build a prototype is made. In the present state of MEMS technology it is used to reduce the number of trial and error physical fabrication iterations required to converge on the final design, masks and processes. In the future, it will also be used to guide the development of improved processes and to assess the sensitivity of the design to normal process variations. It can also play a role in developing design rules.

Finding: While a “clean separation” between design and fabrication is one ideal, the current state of MEMS developments will greatly benefit from accurate, efficient simulation of fabrication processes. The primary benefit will be a reduction in the number of prototype cycles required.

Recommendation: Evaluate process simulators currently available (both for VLSI and MEMS) and support research efforts to advance the state of fabrication process simulation, including continuous-time simulations, process variations, second- and third-order fabrication effects, and performance metrics.

Finding: Process simulation supports fabricators in building a science underlying their empirical results and it supports designers working on new processes who are seldom able to use detailed formulations for specific regions of design space.

Recommendation: Support the basic science underlying the simulation of classes of processes rather than the definition of detailed empirical relations in any narrow process region. Support process simulation capability and process sensitivity studies for the industry standard processes where there can be rapid prototyping to move the technology ahead.

2.5 Infrastructure Summary

Finding: A national infrastructure for design and fabrication of VLSI devices played a key role in building a VLSI community.

Recommendation: Create a national infrastructure for design and manufacture of MEMS, including a set of standard MEMS processes and a MEMS implementation service (MEMSIS), so that a wider community of MEMS designers can be created.

Finding: The rapid dissemination of MEMS design and fabrication knowledge will primarily be achieved by the movement of students from classrooms and university laboratories to commercial developers of MEMS. This movement can be greatly facilitated by increasing the number of students who are exposed to structured design methods for MEMS.

Recommendation: Develop courses and curriculum materials for MEMS design and fabrication. Courses should be created specifically for faculty members and an annual Workshop with participation of both industry and academia to stimulate future research and facilitate keeping pace in curriculum development with developments in the MEMS area.

2.6 Conclusions

(Semi-)automated synthesis of at least some classes of MEMS devices appears to be a realistic near-term result of research into structured design methods for MEMS. This work will draw heavily on the prior developments in digital VLSI, but will also clearly depart in several significant ways. However, the need for a widely-used representation format (language); the clear value of libraries of reusable elements, and the need of efficient, high-quality simulations are all elements in common with the prior developments in structured design methods for digital VLSI.

These elements may take the form of a language to facilitate structured design methods (perhaps an HDL-like language), or may consist of libraries of successful (sub-)elements or devices (although the language/representation that will be used to store these prior designs in the library remains a research issue), or may be analytical methodologies that can rapidly transform a desired functional description into a the description of a physical device (or perhaps into a description of the instructions (*e.g.*, mask-layout) and other processing instructions to create the device), or may be methods to efficiently and rapidly explore the highly complex design space. Languages, libraries and simulation will form the basis for creating a "clean separation" between design and fabrication of MEMS. Research into all of these approaches should be pursued.

The goal of all of these approaches is to free the MEMS designer from the necessity of intimate knowledge of the details of each fabrication process. This approach to creating a "clean separation" between design and fabrication will not only greatly enlarge the community of MEMS designers, and decrease the time and number of prototypes required for each new MEMS device, and increase the quality of MEMS designs (by increasing their robustness to uncontrolled variations), but will also free the fabrication process developers to (more) independently improve their processes.

References

[1] New Paradigms for Manufacturing. Workshop Report NSF 94-123, National Science Foundation, Computer and Information Science and Engineering Directorate, Microelectronics Information Processing Systems Division, Arlington, VA, May 1994. A. Mukherjee and J. Hilibrand, editors.

[2] NSF Workshop on Design Methodologies for Solid Freeform Fabrication. Workshop report, National Science Foundation, Computer and Information Science and Engineering Directorate, Microelectronics Information Processing Systems Division, Engineering Design Research Center, Carnegie Mellon University, Pittsburgh, PA, June 5-6 1995. D. P. Siewiorek, editor.

[3] Small Machines, Large Opportunities: A Report on the Emerging Field of Microdynamics. Report of the NSF Workshop on Microelectromechanical Systems Research, National Science Foundation, Salt Lake city, UT, Hyannis MA, and Princeton, NJ, July 27, and November 12, 1987, and January 28-29, 1988. K. Gabriel, J. Jarvis and W. Trimmer, editors.

[4] BRYZEK, J., PETERSEN, K. E., AND MCCULLEY, W. Micromachines made of silicon. *IEEE Spectrum 31*, 5 (May 1994), 20–31.

[5] MEAD, C., AND CONWAY, L. *Introduction to VLSI Systems*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1980.

[6] PETERSEN, K. E. Silicon as a mechanical material. *Proceedings of the IEEE 79*, 8 (May 1982), 420–457.

3 Workshop Discussion Group Reports

Each of the four discussion groups deliberated over the course of three days of meetings, and produced a report. The objective of each group's report is to briefly summarize the state of the art, the current needs, and articulate a set of research issues and goals.

Additionally, during the discussions with all four groups together, it was unanimously agreed that some elements of infrastructure should also be developed, notably readily available standard fabrication processes, and widely disseminated teaching materials. Thus a fifth group report is included here, briefly addressing the issues of infrastructure.

3.1 Synthesis of MEMS

3.1.1 Introduction

In recent years, computer-aided design tools for the synthesis of VLSI integrated circuits have been a topic of research and development. Synthesis strives to proceed from a high-level behavioral or structural description of a VLSI system down to low-level mask geometry. The various VLSI synthesis levels build upon the mask geometry layout editors that are the designer's interface to the fabrication process.

Arguably, "structured-custom" synthesis paradigms in digital VLSI (hybrid gate arrays, standard cells, chip assembly, parameterizable cores) and rapid prototyping methodologies (field programmable logic and interconnect devices) are "success stories". There has been greater progress in synthesis tools for digital VLSI systems than for analog VLSI systems. The analogy of microelectromechanical systems (MEMS) to analog VLSI may be more appropriate than an analogy to digital VLSI. Many position papers at the May 1994 NSF *New Paradigms for Manufacturing* workshop [1] noted the limits of the analogy between VLSI systems and mechanical systems, particularly with respect to the design process, and implicitly with respect to the prospects for developing "synthesis" tools for mechanical systems.

Our own general session and breakout discussions, while specific to microelectromechanical systems, have recapitulated the general findings of previous workshop attendees:

(i) VLSI has a "complete basis" (the NOR gate), while MEMS does not;

(ii) VLSI has a "clean separation" between function/design and fabrication/process, while MEMS does not;

(iii) VLSI has mathematical constructs (Boolean algebra, correct-by-construction synthesis, etc.), while MEMS does not, and

(iv) VLSI operates on (digital) information and has no moving parts, while MEMS operates in multiple coupled energy domains and has mechanical structures.

Recent research in MEMS has focused on inertial sensors, micro-optics, micro-robots for assembly, and micro-manipulation. The richness of possible applications, and the complexity of contemplated systems (*e.g.*, micro-robots), suggest a great deal of potential for synthesis technology for MEMS. Analogous to the situation in VLSI, synthesis tools for MEMS must operate on several views of the design:

(i) from desired 3-D geometry (device cross-sections) to process and mask specifications;

(ii) from electro-mechanical circuit to component geometry specification; and

(iii) from high-level, functional description of the hardware to electro-mechanical circuit specification.

Overview of VLSI Synthesis Synthesis, in the VLSI context, connotes “something other than custom or hand-crafted”. Today, elements of the VLSI synthesis process contain a number of different design levels and a collection of CAD tools to convert from level to level. At the highest level are Hardware Description Languages (HDL). These languages can include a behavioral and/or functional description of the system. The various hierarchical levels of abstraction correlate well with stages in a standard “design flow”.

From the HDL, a synthesis system would contain a CAD tool with algorithms which can apply transformations to produce an intermediate design representation. This intermediate design representation may be at the control-data flow graph or Boolean network levels. Another CAD tool may then input the control-data flow graph and search for area and time design trade-offs within the space of feasible (*i.e.*, correct) designs. This design trade-off analysis tool would output a revised list of hardware resources and a control “schedule” in a structure description language format.

At the next lower level, algorithms which map the revised intermediate design representation to a library of component “macros” (RTL-level blocks, standard-cell, etc.), would be invoked. This process is essentially performed via a covering formulation. At the lowest level, there will be algorithms which embed the physical design and its connection topology onto silicon resources. A number of reasonably well-established CAD tools will handle the placement and routing tasks and produce a final CIF or GDS II mask geometry file if the design is to be produced via a typical CMOS process.

Issues for MEMS Synthesis From the above discussion of a typical VLSI synthesis system, a number of similarities and differences exist when trying to propose a synthesis systems for MEMS. These include:

- Mechanical coupling and multi-functional structure complicate the physical embedding task. “Back-loading effects” and a host of constraints, along with a possibly richer notion of “component library” that spans parameterized generators and multiple fabrication processes, complicate the technology mapping task. Fundamental research must address:

  (i) new means of representing/parameterizing the design space (*e.g.*, at the “circuit” and “component” levels),

  (ii) new means of abstracting optimizable objectives from the design description, the design constraints, the “netlist” of selected components, etc., and

  (iii) new means of performing constrained/heuristic optimization and heuristic search.

- The semantics of interface specifications for components requires development (*e.g.*, semantics of a component structure in terms of required adjacency to X, shielding property for Y, barrier for flow of Z, etc.). There is a need for engines and algorithms that can reason about the increased complexity and variety of interconnections that are possible in MEMS systems.

- Parameterized component libraries and library generators require further development. For example, generalized semantics for terminal locations, internal (to the component) connections, process technologies, technology scaling, drawn and process dimensions, various performance parameters (time constants, etc.) need to be defined and standardized.
- There are differences in mapping geometry to process and mask for MEMS as opposed to VLSI.
- In VLSI, designs can be reduced to a simple set of library logic cells, such as NAND and NOR. A basic cell library simplifies design reuse. In MEMS, it is more difficult to propose a basis set, from which one can combine into a large class of systems.
- An interesting long-term goal would be the development of a "complete" or "admissible" search and optimization methodology for "optimal design". Possible directions include a generalized theory of transduction, or the notion of a fundamental building block for MEMS (*e.g.*, a Cartesian block with inertia, damping, spring constant, coefficient of thermal expansion, etc.).

3.1.2 Findings and Recommendations

Synthesis is complementary to analysis which includes the tasks of extraction and simulation. Accompanying each step from a given level to the next higher level is an extraction, whereby the output of analysis at one level can be used for analysis at the next higher level. It is important to have sufficient analysis tools so that goals such as the following can be met successfully:

(i) top-down design from a functional or other high-level specification, or

(ii) iterative design that includes semi-automatic exploration of the design space and achieves short design cycles.

The following summary lists suggested directions and motivations for several broad levels at which synthesis for MEMS should be developed. To organize the findings and recommendations, Table 1 shows the different "levels" of MEMS design. The synthesis process involves traversing this figure from the top system level down to the bottom process-mask level. Conversely, analysis involves traversing this figure from the bottom up. Each of the subsections below relates to the tasks needed to proceed from one level of the table to the next level below.

Finding: MEMS Hardware description language. At present, a HDL for MEMS is not available. Several characteristics of a potential language were considered. For example, the HDL may have a "C" language or MATLAB style syntax. The language syntax should help to facilitate the mapping onto the building blocks in the MEMS cell library.

Synthesis Level	3 Year goal	10-Year goal
SYSTEM	Lang. requirements	HDL language
Model including (dynamics)	HDL to schematic synthesis	Formal design and verification methods
	Diagnostic structures	Resource allocation
		Packaging, assembly and identification
MULTI-DOMAIN SCHEMATIC	2D mech-electrical	Chemical, magnetic fluidic and thermal Other domains
	Library: actuator, spring, mass, damper elements	Place & route cues
		Shape synthesis
COMPONENT	Tool prototype	3D shape generator
Physical shape (structure)	2D shape generator (homogenization)	Yield/cost analysis tools
	Layout generator with component library for fixed processes	Process variation compensation tools
Process compiler (MISTIC)	Implement & distribute process compiler such as Tanner, CaMEL	
PROCESS — MASK	Extensions to bulk micromachining mask/layout	Assembly sequence synthesis
	Develop process library technology file	Reliability
	DRC & extraction for specific processes	

Table 1: Goals for Synthesis Research from System Issues down to Process and Mask. Synthesis Proceeds Downward ⇓ while Analysis Proceeds Upward ⇑.

Recommendations: Define a MEMS Hardware Description Language.

- Develop a Language Requirements Manual (LRM) and perhaps a draft specification. [3-years]
- Incorporate a description of multiple energy domains, transduction, and packaging into the HDL. Describe the physical/spatial relationships between objects. Investigate the applicability of VHDL-A, (analog extension to VHDL). Provide information on MEMS needs and requirements (*e.g.*, distributed /non-lumped systems, or thermal expansion) to the committee developing the VHDL-A specification. [5-years]
- Proceed with HDL Language standardization and encourage widespread use. [10-years]

Finding: HDL to "Multi-domain Schematic". Automated synthesis from HDL is currently not available. The problem of synthesis can be further divided into synthesis of systems and synthesis of custom components. The high level description of a system can be used to assemble the system from a library of existing components. However, the library may not have the necessary components for all systems. Thus it is also necessary to synthesize custom components from a functionality specification. This can be accomplished by determining the device topology needed, then scaling the dimensions appropriately and if necessary extracting the process required to fabricate it. Synthesis is necessary because the bottom up approach takes too long to design.

For the system designer, the goal of synthesis is not necessarily designing the optimum device but is rapid prototyping and "design reuse" through component libraries. For the custom component designer, the goal is maximum performance. These two goals may lead to different synthesis pathways.

Recommendations:

1. Define schematic symbols, interfaces and connection rules for MEMS components. The near term focus is on planar technology, and on thermal, solid mechanical, and electrical domains. [3-years]

2. Develop an HDL-MEMS-to-schematic translator. This tool would provide a graphical interface to the MEMS schematic symbols (analogous to a VLSI schematic capture tool). [3-years]

3. Study the Bond-Graph representation, which is based on power flow. This representation provides a framework for systems involving several energy domains. It not only provides a common representation for all elements belonging to different energy domains, it also permits couplings among different domains. It works on the principle that elements/subsystems of every domain can be broken into three types of common elements such as storage elements, dissipators, and energy sources/sinks.

There are system level simulators that read a bond-graph representation and simulate the behavior of the system (*e.g.*, CAMAS from the University of Twente, Netherlands and ENPORT from Michigan State University). Determine how well related electronic elements fit the bond-graph representation, that is, how well can we describe the electronic elements (*e.g.*, transistor, Op-Amp) in terms of storage-dissipator models. Explore the bond-graph approach as a means to obtain a system level description language integrating mechanical and electrical components. [3-years]

4. Formal verification and "correct-by-construction" design methodologies. [10-years]

5. Develop tools for MEMS "signal" flowgraph generation, scheduling algorithms (parallel/serial, area time trade-offs), and automatic resource allocation (number of components). Develop tools with knowledge of constraints on signals, and binding of resources (operations to components). These tools would be analogous to similar steps in VLSI high-level synthesis. [10-years]

6. Provide support tools for packaging and assembly, analogous to tools for VLSI padframe generation. [10-years]

Finding: "Multi-domain Schematic" to 3-D Shapes. In a "strawhorse" example, the input is a multi-domain schematic. For example, consider the design of a resonator composed of a spring and a mass, shown photographically in Figure 3 and schematically in Figure 4. The system has intrinsic damping and is driven by an electrostatic or electrothermal actuator. The terminals can have associated vectors of energy domains, or be specific to a particular energy domain. Design tools would then generate component layout. Tanner Research Tools and MCNC's CaMEL efforts are the current state of the art in terms of parameterized layout generators.

Determine the shape of the component from the desired behavior. Use techniques like homogenization (3-10 years) at the electromechanical circuit level. The motivators include: design re-use, formalized repository or mechanism for reproduction of design know-how. Complications include: mechanical coupling, multiple interacting energy domains, etc.

Recommendations:

1. Develop a multi-domain schematic representation which includes actuators, springs, masses and dampers. Develop extended semantics (cf. the observations of "multifunctionality" in earlier workshops). For example, a link in VLSI denotes electrical connectivity, but often (*e.g.*, at the place-and-route level) does not even have associated information giving the direction of signal flow. In contrast, a link for MEMS might denote impermeability to a fluid, heat conduction path, rigid (straight-line shape) mechanical coupling, etc.

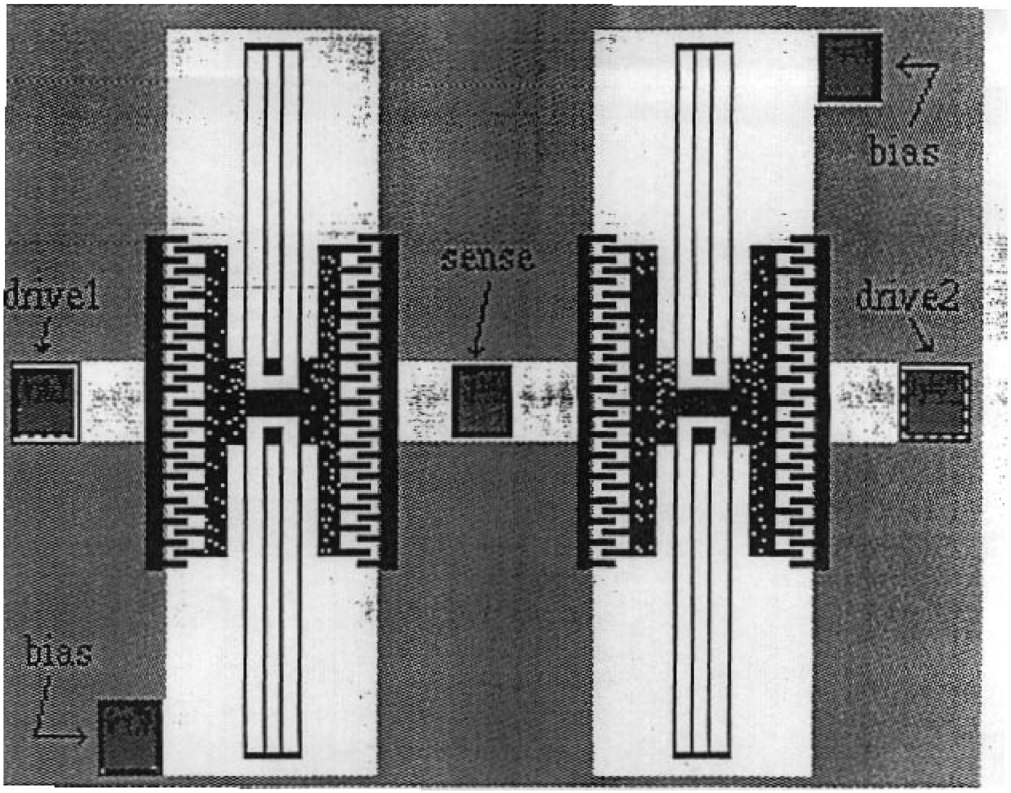


Figure 3: Photograph of a MEMS Resonator.

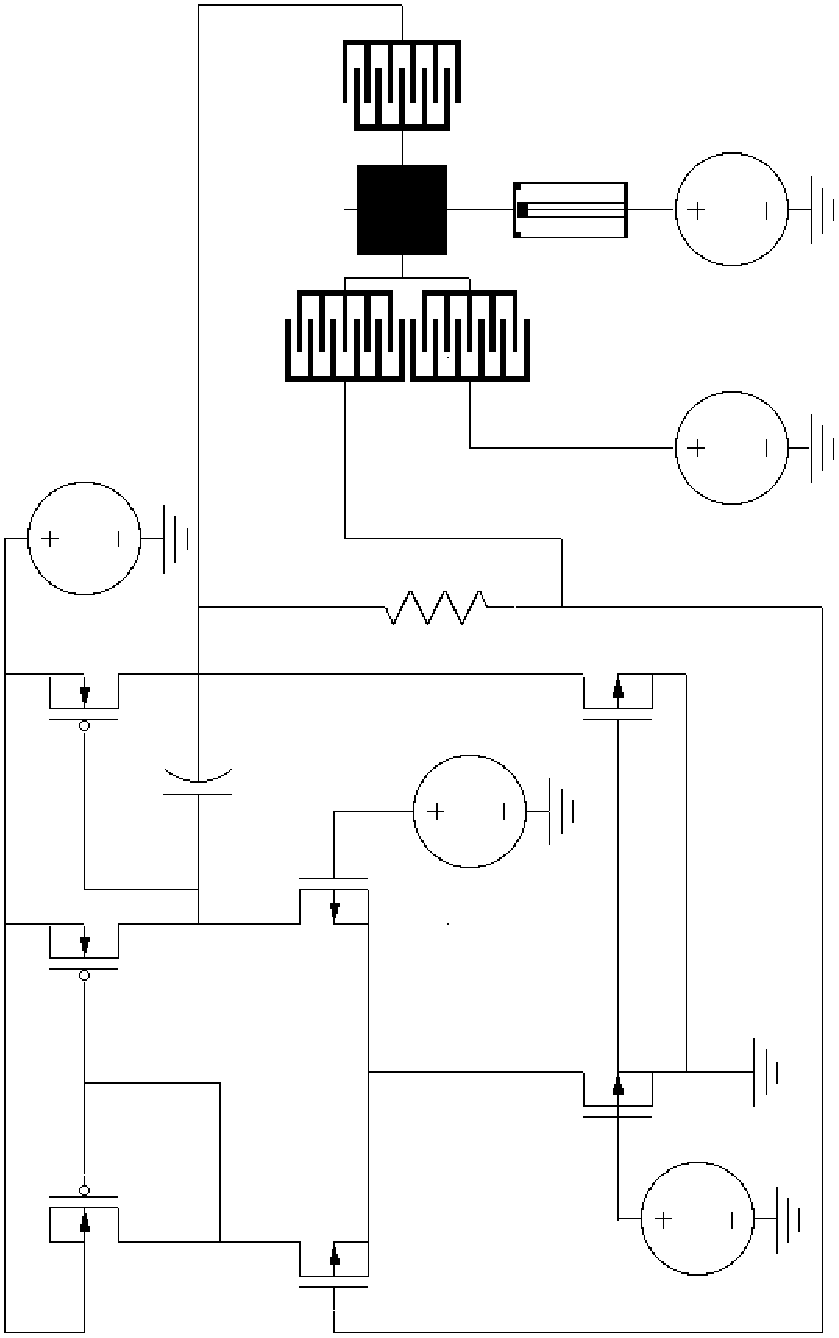


Figure 4: MEMS Resonator Schematic.

One approach might incorporate a generalized vector of potentials (*e.g.*, temperature, force, electrostatic potential, pressure). Interconnection links would then have semantics that included flow-like quantities (*e.g.*, heat, displacement, current, fluid). This would offer the possibility of straightforward paths to simulation tools like SPICE. [5-years]

2. Develop prototype tools for the following:

    - Develop shape generation: For example, given a mass and spring constant, generate a flexure, support structure, etc.
    - Develop "Search mechanisms" for extracting library elements.
    - Develop place and route tools to complete the schematic. Place and route is essentially an optimization step. The problem is one of physical embedding under tremendous constraints. Both the technology mapping and the physical layout will require significant departure from current VLSI approaches. There will typically be a very discontinuous "feasible region" for the design, and steps in the flow may actually be quite iterative. Technology mapping, parameterized component generation, place-and-route might together constitute a loop. [3-years]
    - Integrate yield, sensitivity analysis (to process variations), and cost issues into process synthesis. [10-years]

Finding: Shape to Process Flow/Mask Geometry. If the process is fixed, then process data and choice of generator from a generator library can also yield mask data for a component. This is the Tanner and CaMEL approach. For CaMEL, there is a generator utility that accesses a generic library of electro-mechanical elements including actuators, gears, simple hinges and accelerometers. The non-parameterized elements are generated using PERL scripts and the parameterized elements are created using compiled programs. There are more non-parameterized elements than parameterized elements. Tanner has developed an applications-level interface to their design database and user-interface capabilities, which can be used to develop parameterized component layouts. Currently a C-language interface is supported. The benefit of the Tanner approach is the wide acceptance of C and the L-Edit program.

Tools that automatically synthesize process flows from cross-sectional specifications are needed for "custom" device synthesis. Generators must output "design rule correct" (*i.e.*, feasible with respect to process) instances. Existing tools from Univ. of Michigan (MISTIC) have made good progress to this end for surface-micromachining. Complementary work in mask layout synthesis for bulk micro-machined structures should be encouraged.

Process perhaps should not be so intimately coupled to function and design. It is possible that more designs are "reachable" via a technology-specific suite of synthesis and exploration tools than are reachable via the current approach (within which process design essentially follows from function). The task of building a significant component library for a particular identified process can spur development of tool infrastructure.

Recommendations:

1. Develop a full implementation of current research software for thin films, for example the MISTIC-type approach. Then extend the implementation to bulk micromachining (analogous mask layout capability for bulk-micromachining) and 3D. Develop library generation tools (correct for any given process). Efficient search over the solution space is critical. Disseminate tools to all users. Reduce algorithm complexity of the implementation. [3-years]

   Both types of generators should be integrated into general purpose layout tools. Continue to develop the CaMEL library. Generate simulation models linked to layout generation. [3-years]

2. Understand use of time etching for component shaping. This is an inverse of recent anisotropic etching simulation work. [3-years]

3. Define and develop technology files for standard processes. [3-years]

4. Develop libraries which are correct-by-construction for any given process. For fixed processes, process data and choice of generator from a generator library can also yield mask data for a component. Develop design rules and design rule checking (DRC) capabilities. Generators must output “design rule correct” (*i.e.*, feasible with respect to process) instances. Emphasize “design reuse.” [3-years]

5. Investigate assembly sequence, and reliability issues. [10-years]

Finding: Synthesis from Performance to Mask Geometry. This process essentially skips the 3D geometrical view. Recent research is exploring this approach. The CaMEL and Tanner systems can generate geometries from physical parameters, but these tools do not include evaluations or simulations of performance.

Recommendations:

1. Develop performance driven layout generation. Generate layout from these three data inputs: material/process data (from handbook, FEA, measurements, Technology File), libraries of parameterized cells (and cell generators) and performance specifications. [3-years]

2. Promote research to understand process flows – including time etches – with respect to the layer structures. Demonstrate component synthesis. [3-years]

3. Demonstrate synthesis of smoothly varying structures and even further with the LIGA technology. [10-years]

3.1.3 Summary

Despite the differences between digital VLSI and MEMS, sufficient parallels exist to recommend a program of research to develop (semi-)automated methods for synthesizing at least some classes of MEMS devices. The differences between digital VLSI and MEMS strongly suggest that structured MEMS synthesis methods will not be a direct outgrowth of VLSI methods, however, many of the underlying constructs (*e.g.*, language, modeling, simulation, etc.) should be similar in philosophy, and appear likely to serve as guides for initial research in this area.

VHDL-A appears to be an appropriate starting place for synthesis language-related research. The development of libraries of previously successful designs is also important, but can proceed (at best) in parallel with synthesis language development. Approaches to transforming a description (perhaps of the function) of a desired device into a description of the physical device (including translation-like methods as well as search methods) have a great deal of promise for synthesis of MEMS devices, and research in this area should be encouraged.

3.2 Simulation of Function

3.2.1 Introduction

CAD tools for MEMS should be developed that are easy to use for novices, electronic circuit designers, and MEMS experts. MEMS "experts" abilities are usually focused on sub-areas of MEMS and they too will greatly benefit from basic CAD tools. There is room for a hierarchy of representations and models of MEMS to satisfy the needs of a spectrum of users. Features of easy to use MEMS CAD tools include:

- integrated toolset with capability to design at different levels and views,
- generation and extraction to different views and levels,
- fast algorithms for multi-domain analysis,
- libraries of reusable elements,
- support for top-down and bottom-up design.

Many design issues govern the CAD representation of MEMS and VLSI frameworks. To see if a VLSI framework or design methodology is extensible to the MEMS field, the differences and similarities between the two must be explored. In VLSI, the major part of the design occurs in a single primary energy domain with interactions described primarily by electrical quantities i and v. In MEMS, the components of the system work in multiple energy domains - electrical, mechanical, thermal, radiant, chemical, magnetic, acoustic and fluidic - with coupling between them. Thus, any CAD framework for MEMS must support representation and analysis in multiple energy domains. In VLSI, a 2D layout abstraction is supported whereas in MEMS a 3D representation of structures is needed by the designer–especially for analysis. Process level design is also necessary in the MEMS arena and requires a separate design/simulation framework. Unlike VLSI design, there currently isn't a "clean separation" between the designer and the fabrication process for all MEMS processes. In addition, digital VLSI design typically has many higher levels of abstraction that are not currently used in MEMS design.

However, there are still many similarities between MEMS and VLSI design methods. The multiple levels and views of MEMS components and system used by designers are similar to those in VLSI. Different levels of abstraction are used for analysis spanning FEA/BEM to lumped element macro-models to functional models. MEMS bears a closer resemblance to MMIC or analog VLSI design where "custom" layout is common. There is a similar need to meet specifications and optimize layout to satisfy design and process constraints.

In order to provide suitable simulations of function to support MEMS design, appropriate design levels and representations are necessary. In addition, sufficient generality must be present to support multiple energy domain systems.

Design Framework

Some ideas can be extracted from the VLSI paradigm to form a design framework for MEMS. VLSI design evolved into a design hierarchy of different levels, as illustrated in Figure 5. High to low levels of abstraction in VLSI include system, register, logic, circuit, device, and materials. At each level, there exists three different views: behavioral, structural, and physical. The behavioral view represents how a design works, the structural view represents how elements are connected together in a design, and the physical view represents how designs are implemented.

The structure of the VLSI design matrix may be applied with modification to formation of a MEMS design hierarchy. The concept of behavioral, structural, and physical views as equally important representations of each hierarchical level can be directly borrowed from the VLSI domain. Levels in our vision of a MEMS design hierarchy are somewhat parallel to those in VLSI, however there are some important differences. An initial partitioning of design levels from the top down includes system, module (sub-system), circuit, device, and material levels. This matrix of MEMS design representations is illustrated in Figure 6 for a prototypical MEMS system. Pictorial illustrations at each level and view are taken from a relatively mature MEMS application, a torsional micro-mirror. The micro-mirror example couples together several energy domains including mechanical, electrical, fluidic, and optical. The example micro-mirror device is a micromechanical plate suspended by two simple beams and electrostatically deflected by a pair of parallel-plate actuators. Individual micro-mirror devices may be incorporated into larger arrays, and these arrays may be part of a larger system.

The system level deals with high-level architecture issues such as component placement, component interaction, upper-level packaging, and data bus routing. Systems could be implemented as a set of discrete parts, as a hybrid package, or as a monolithic microsystem. This highest level of abstraction in MEMS is not well understood and will be better defined as individual MEMS devices and processes mature. A "strawhorse" concept at this level is illustrated in Figure 6. The physical view is a drawing of the packaged system that specifies the physical placement and size of its constitutive parts. The structural view specifies the high-level component interconnections, which includes coupling between multiple energy domains. The behavioral view includes many kinds of microsystem analyses, yet undefined, dealing with topological, architectural, and specification issues.

The module level deals with sets of components that combine in a sub-system to satisfy an individual function. The specific example in Figure 6 is a micro-mirror array that provides a display function. The physical view displays the layout between micro-mirror devices, the structural view identifies the multi-domain interconnect between devices, and the behavioral view includes system analysis of the array. The specific needs for tools at this level may overlap with the requirements from the system and circuit levels. Unification of the various energy domains is one target area that could significantly impact usefulness of MEMS representations at both the module and system level. Such unification would enforce energy conservation of the behavioral representations, which is of fundamental importance.

The circuit level involves representation of individual sensor and/or actuator compo-

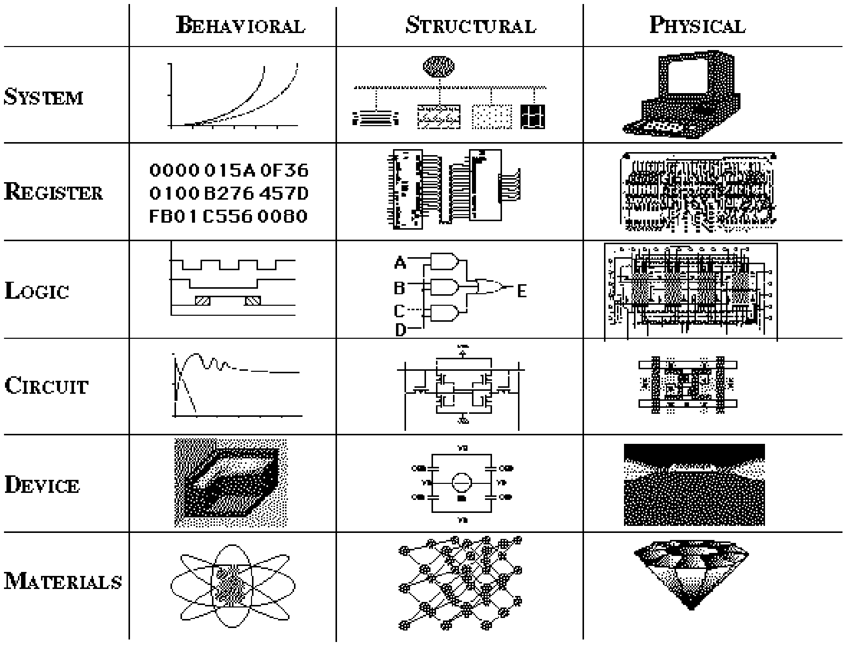


Figure 5: Matrix of VLSI Design Representations.

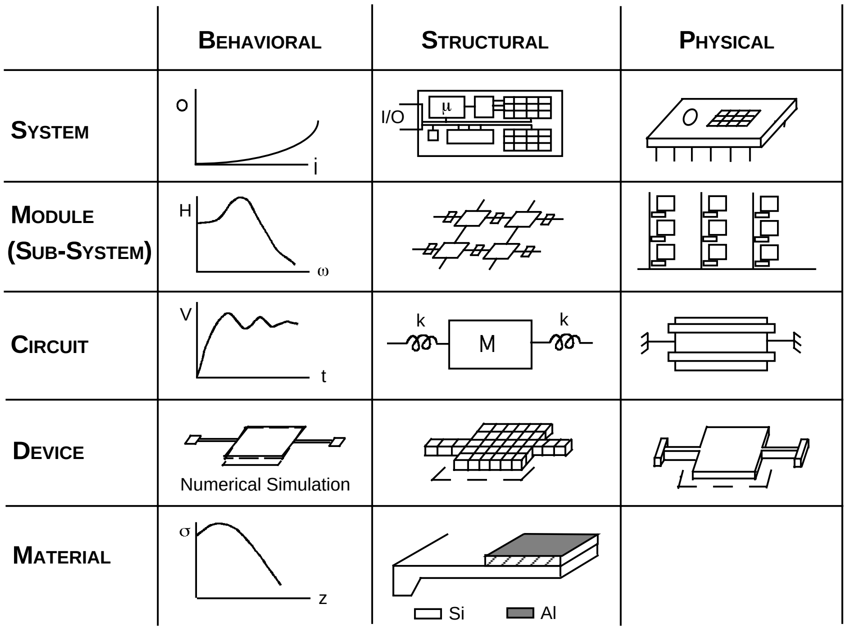


Figure 6: Matrix of MEMS Design Representations.

nents. The physical view is device layout, which could be a 2-D, 2.5-D, or 3-D representation. The structural view is envisioned as a multi-domain MEMS schematic, specifying the connectivity between lumped-parameter MEMS elements. For the torsional micro-mirror example, the schematic representation specifies the connectivity of several MEMS lumped-parameter elements: a plate mass which acts as the mirror, two beam springs, and two electrostatic actuators. A schematic view at the circuit level does not presently exist for MEMS. Such a view may be an important intermediate representation for MEMS synthesis and analysis tools. The behavioral representation includes multi-energy-domain mixed-signal simulation in both the time-domain and frequency-domain.

The device level primarily deals with numerical representations of the MEMS device. The physical view is a full 3-D model of the device. The structural view is a discretized version of the 3-D model, with enough detail to extract the essential dynamic and static modes of the device. The behavioral view primarily involves numerical simulation, such as finite-element and boundary element analysis, with coupling between energy domains. Research needs include fast numerical analysis of large multi-domain problems. Making numerical analysis tools “MEMS-friendly” is important in order to improve accessibility of this technology.

The lowest level in the hierarchy is the material level. The distinction between the physical and structural levels is still unclear at this level. These views include a 3-D process view of the device showing connection between different materials. The behavioral view deals with process simulation and modeling. The simulations would include materials constitutive relationships at all relevant energy domains, such as predicting strain gradient in a material.

Several capabilities are required in the design representation to support movement between the different views and levels of hierarchy. This list is not meant to be comprehensive, but suggests some of the critical information required in the database which supports the design hierarchy. Some capabilities required of the design representation include:

- a 2-D representation (similar to an electronic schematic or a mask layout),
- a 3-D representation (perhaps a 3-D solid/geometric model),
- 2-D and 3-D meshes (finite element discretization (for stress, strain, electric charge, etc.))
- netlist from schematic (the ability to determine function from a schematic),
- netlist from layout (the ability to determine function from a layout),
- material properties,
- links to libraries (to indicate which models, model parameters, layout and schematic cells are used),
- specifications for the system

- layer database (to indicate which layers are possible, permissible order of layers, etc.),
- design rules (conservative rules for minimum device spacing, etc.),
- simulation output (field representations and graphs).

3.2.2 Findings and Recommendations

Finding: Specification. An extensible specification is needed for the design framework, possibly based on Figure 6.

Recommendation: Explore existing VLSI design/CAD system architecture(s). Develop an architecture for an integrated CAD environment for MEMS.

Finding: Libraries. Representations, interchange formats and standards must be defined so that libraries may be exchanged, possibly over the internet between designers. Libraries must be built encompassing all levels of the design hierarchy and for all the views: schematic, layout and behavioral.

Recommendation: Explore existing VLSI design/CAD library structure. Develop an extensible library format for exchange of MEMS designs.

Finding: Analysis of Multiple Energy Domains. Since MEMS incorporates multiple energy domains (and VLSI does not), analysis capabilities are needed to efficiently simulate function in these domains, including interactions (*e.g.*, application of stress alters shape which alters the distribution of static-charge, which alters the stress, etc.).

Recommendation: Develop a theory of multiple energy domain analysis and an implementation sufficiently efficient to utilize during design.

Finding: Design Tools. There are many tools that can accelerate MEMS design and be utilized in a design framework.

Recommendations: Initiate research to develop the following:

- Layout verification tools: MEMS specific design rule checkers, component extractors and layout vs. schematic netlist comparers.
- A hierarchical schematic editor that supports MEMS specific information.
- Faster and coupled multi-domain simulators for device level simulation: in the short term possibly electrical, mechanical, magnetic, micro fluids and thermal.
- Circuit level simulators possibly based on a SPICE like paradigm.
- Tools that can extract model parameters from simulation outputs or physical representations.
- Tools that build macro-models from simulation results.

- MEMS specific solid modelers.
- A tool that helps designers choose among high level design alternatives.
- Choice of hardware modeling languages: possibly VHDL-A.

Finding: Long-term Goals. In 5-10 years, MEMS designers will utilize the highest levels of the design framework as more complicated system level designs are undertaken.

Recommendations: To support more complicated system-level MEMS design work, the following elements should be developed:

- A more complete design framework implementation encompassing all views and levels of design hierarchy.
- A more complete implementation of the mixed regime and the unified theory of multi domain analysis and design.
- The following tools may be useful at the 10 year time scale as MEMS designs become more complex:
  - A place and route tool,
  - Extraction and synthesis tools at all levels of design hierarchy.
- A MEMS version of MOSIS.
- Optimization to be supported by the design libraries.
- Robust design using statistical analysis of process variances.

3.2.3 Summary

Rapid, efficient, accurate, analysis of many alternatives during the design process is crucial to developing high-quality, robust devices and systems. Developing a widely-used, extensible framework for design analysis, perhaps building on the approach used in VLSI analysis (switch-level simulations, netlist extraction, etc.), will significantly shorten the time, and reduce the cost, of MEMS design.

3.3 Digital Data Interchange Languages

3.3.1 Introduction

Standard digital data interchange representations (*e.g.,* Caltech Intermediate Form (CIF)) have had a beneficial effect on the development of structured design methods for VLSI. It enabled the development of design tools to address issues at all levels of abstraction, and provided access to technology that was otherwise unavailable to a wide community of electronics designers. The resulting innovations ranged from powerful new digital design tools, to computer architectures that spawned new corporations. The simplified design tools in turn made VLSI design accessible to all university students interested in computer science and electrical engineering. The standard interface to manufacturing produced an economy of scale that could not be achieved by individual researchers or educators.

Some of the lessons that were learned in VLSI might be beneficially applied to the MEMS community. An examination of a typical design and manufacturing flow in mechanical systems reveals certain similarities in data interfaces between major modules where the VLSI approach might be emulated. Figure 7 illustrates a typical mechanical system design process. However, the mixture of mechanical and electronic components and functions will require new research to develop the multi-domain, multi-dimensional representations required.

Five specific areas where research work on standard representations will benefit the development of MEMS are:

1. 3-D solid modeling representation;
2. Interfaces between design tools (*e.g.*, between 3-D modeling, analysis, and process design);
3. Representations for mask-layout, process specification, and metrology;
4. Interfaces between design and fabrication (*e.g.*, between process design tools and fabrication);
5. Representation for function (*e.g.*, the MEMS equivalent of digital VLSI's Boolean logic).

3.3.2 Findings and Recommendations

Common data representations which are widely used are critical to rapid innovation in MEMS.

Finding: In MEMS it is important to describe the design in such a way as to maintain a "clean separation" between the design and fabrication processes in order to achieve the benefits of a structured design methodology. This requires that the designers describe the desired object in a well-defined language that is easy to learn and that is

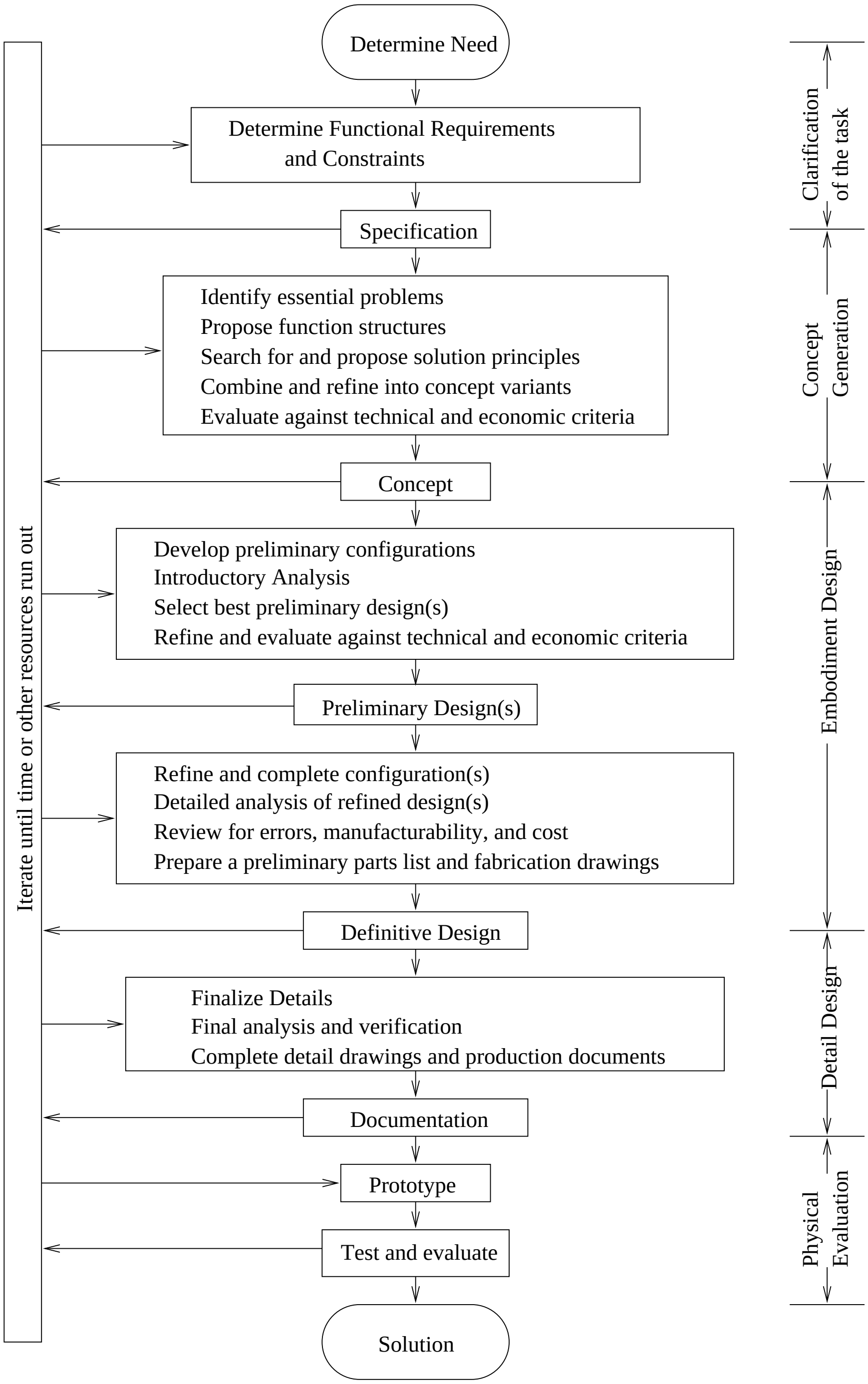

Determine Need
Determine Functional Requirements and Constraints
Specification
Identify essential problems
Propose function structures
Search for and propose solution principles
Combine and refine into concept variants
Evaluate against technical and economic criteria
Concept
Develop preliminary configurations
Introductory Analysis
Select best preliminary design(s)
Refine and evaluate against technical and economic criteria
Preliminary Design(s)
Refine and complete configuration(s)
Detailed analysis of refined design(s)
Review for errors, manufacturability, and cost
Prepare a preliminary parts list and fabrication drawings
Definitive Design
Finalize Details
Final analysis and verification
Complete detail drawings and production documents
Documentation
Prototype
Test and evaluate
Solution
Iterate until time or other resources run out
Clarification of the task
Concept Generation
Embodiment Design
Detail Design
Physical Evaluation

Figure 7: Mechanical Design Process Flowchart.

suited to hierarchical descriptions, while the fabrication community takes responsibility for conversion of that description to masks used to build the desired object. Standard processes, whose capabilities (design rules) are known in the design and fabrication communities, can facilitate this separation.

Recommendation: Research efforts should be focused on design methodologies which support the "clean separation" of design and fabrication. That is, well understood standard processes with proven design rules should be supported with design hierarchies and design tools which encourage the designer to describe the layers of the final object resulting from the fabrication process rather than the mask set.

Finding: 3-D Solid Modeling. Existing 3-D solid modelers currently used for macro-mechanical system design may not be adequate and appropriate for MEMS.

Recommendation: Identify and adopt a standard for data representation of 3-D solid models for interchange between design tools. [3-years]

Finding: Mask-layout. Two clear areas for extension of currently available mask-layout software are: an improved capability to define non-polygonal geometry; and a mechanism for interaction between the designer and the pattern generation system to produce an optimal match.

Recommendation: Develop and adopt extensions to CIF or GDSII to accommodate non-polygonal geometries in order to provide a more efficient interface with pattern generation equipment. [3-years]

Finding: Fabrication Process Language. In automated (CNC) mechanical fabrication, a standard language (G-Codes) is nearly universally used. In semiconductor manufacturing, MAP is commonly used. An extensible, hierarchical fabrication process specification is needed for MEMS (including flexibility in defining the materials and layers of the final product) to provide designers and fabricators with a common language for communicating manufacturing instructions.

Recommendations:

- Develop and adopt a standard for the specification of simple sequences of unit processes sufficient for simulation and fabrication of surface machining. [3-years]
- Develop and adopt a standard for the specification of simple sequences of unit processes sufficient for simulation and fabrication for detail within unit processes. [5-years]

Finding: Metrology. It is critical to assure that MEMS fabrication processes have been properly carried out for a given design. This will aid in debugging a new development by determining whether a failure is due to mis-fabrication, or a design error. Test structures that are implemented for this purpose will be highly sensitive to process variations and are easily measured at the end of the process. In addition, a second use of these test structures provide model parameters for designers who may reuse the same process steps.

Recommendations:

- Identify and document existing process monitors of use to the MEMS community and make available over the WWW. [1-year]
- Develop verify and make available test structures, procedures and programs to quickly determine process quality, such as endpoint completion in MEMS processes. Provide a metric of technical progress in order to have a benchmark to measure improvement over time. [3-years]
- Work with the industry to agree upon standards for test structures and test methodologies for MEMS processes. Identify opportunities for in-process monitors to more quickly evaluate the progress of MEMS process steps. [5-years]
- Develop sufficient understanding of the methodology and materials to predict reliability of MEMS components and to provide tools for improving materials technology for maximum lifetime. [10-years]

Finding: Language for Function. Digital VLSI utilizes Boolean logic to describe function. The existence of a formal language for function has many advantages, among them are the ability to represent and analyze function mathematically. Macro-mechanical systems, at present, has no single language for function, but instead utilizes many different representations for the many different energy domains. Currently MEMS also lacks a unified representation of function.

Recommendation: Develop a representation for MEMS function, perhaps based on Bond-Graphs. This representation provides a framework for systems involving several energy domains. It not only provides a common representation for elements belonging to different energy domains, it also permits couplings among different domains.

3.3.3 Mask Layout Geometry vs. Desired 3-D Shape

In VLSI, currently, the mask describes the final desired (2-D) geometry. The fabricators take into account the details of their fabrication process and pre-distort the mask so that once fabricated, the desired shape emerges. This establishes the “clean separation” between design and fabrication.

In MEMS, currently, the required pre-distortion is geometrically complex (due to the 3-D nature of most MEMS) and generally unknown. Therefore MEMS mask layout geometry is used as the mask directly. The required pre-distortion (compensation, etc.) must be determined by the MEMS designer, generally over many prototype cycles and with much experimentation. Furthermore, this experimentation does not usually, at present, result in a generalized understanding that can be applied to different MEMS devices.

At issue is whether the MEMS designer can be (or should be) insulated from the details of the fabrication processes. If this is desirable and/or possible, then the geometry that is communicated to the fabricator should be the desired (3-D) geometry. Conversely, if the relationship between mask layout geometry and final shape is too complex, MEMS designer must have available accurate fabrication simulations so that the required pre-distortion of the mask can be determined with the smallest number of fabricated prototypes possible.

Finding: It appears possible to maintain a clean separation between design and fabrication for some classes of MEMS devices. Establishing this clean separation, where possible, appears to be the most direct route to achieving the benefits of a structured design methodology for MEMS.

Recommendation: To broaden these classes and encourage the practice outside these classes, research efforts should be focused on design methodologies which support the clean separation of design and fabrication. That is, well understood standard processes with proven design rules should be supported with design hierarchies and design tools which encourage the designer to describe the attributes of the final object resulting from the fabrication process rather than the mask set.

While the challenges in developing structured design methods for MEMS that preserve the "clean separation" are significant, the benefits of such methods will greatly enhance MEMS developments in those areas where the design and fabrication communities can make use of such methods.

3.3.4 Summary

Much about the historical development of digital VLSI suggests that a common interchange language (CIF) greatly facilitated the rapid advances. While MEMS developers commonly use CIF to describe mask-layout geometry, CIF lacks many elements needed for a complete data interchange language. The development of a common interchange language for MEMS is likely to play a pivotal role in accelerating the advances in MEMS.

3.4 Simulation of Fabrication Processes

3.4.1 Introduction

Process simulation is the conversion of two dimensional mask layout geometry and process information to a three dimensional representation of an output shape for the purposes of visualization and FEM and other CAD tools. MEMS research currently involves a large number of different processes and structures, and there is a need for accurate and flexible process simulators. State of the art MEMS design involves several design cycle iterations; the incorporation of process simulators in rapid design for MEMS would reduce the number and cost of design iterations. Even more than VLSI, the growth of MEMS will be dependent on accurate process simulations.

Process Simulation Tools

While VLSI process simulation tools are available, their generality and applicability to MEMS process simulation has not been fully explored. Existing VLSI tools have not yet been fully integrated in MEMS CAD systems. Existing tools can be divided into three broad groups: inexpensive basic tools, high end commercial systems, and university research/experimental systems. A study of existing process simulators should be undertaken to evaluate their applicability to MEMS processes. These should be extended to incorporate second- and third-order effects (mask undercutting, etc.).

Level of Detail of Simulation

Both VLSI and traditional macro-mechanical design have an established knowledge base of acceptable engineering approximations for different engineering domains, *i.e.*, rules of thumb. How detailed spatially and temporally, must a MEMS process simulation be? MEMS design tools need to be developed that allow the designer to easily specify the level of approximation of simulation for their designs. Note that different structures within a design and different energy domains (*e.g.*, thermal vs. mechanical) may have different levels of detail. Additionally, design techniques need to be developed in the MEMS community that guide the user in selecting the appropriate level of detail. These techniques may be application dependent. Basic design tools are needed in the immediate to short-term future, with greater flexibility to follow on a longer time scale.

Type of Simulators

The complexity and cost of process simulations can vary widely. What distinguishes them is the amount of separation between designer and fabrication process. Novice designers may have little need or desire to know the details of the fabrication process, while experts may need extensive knowledge. The type of simulator a user requires depends on whether the designer is concerned with standardized systems or custom systems. Thus there may be two broad categories in the need for process simulators: an inexpensive basic simulator which

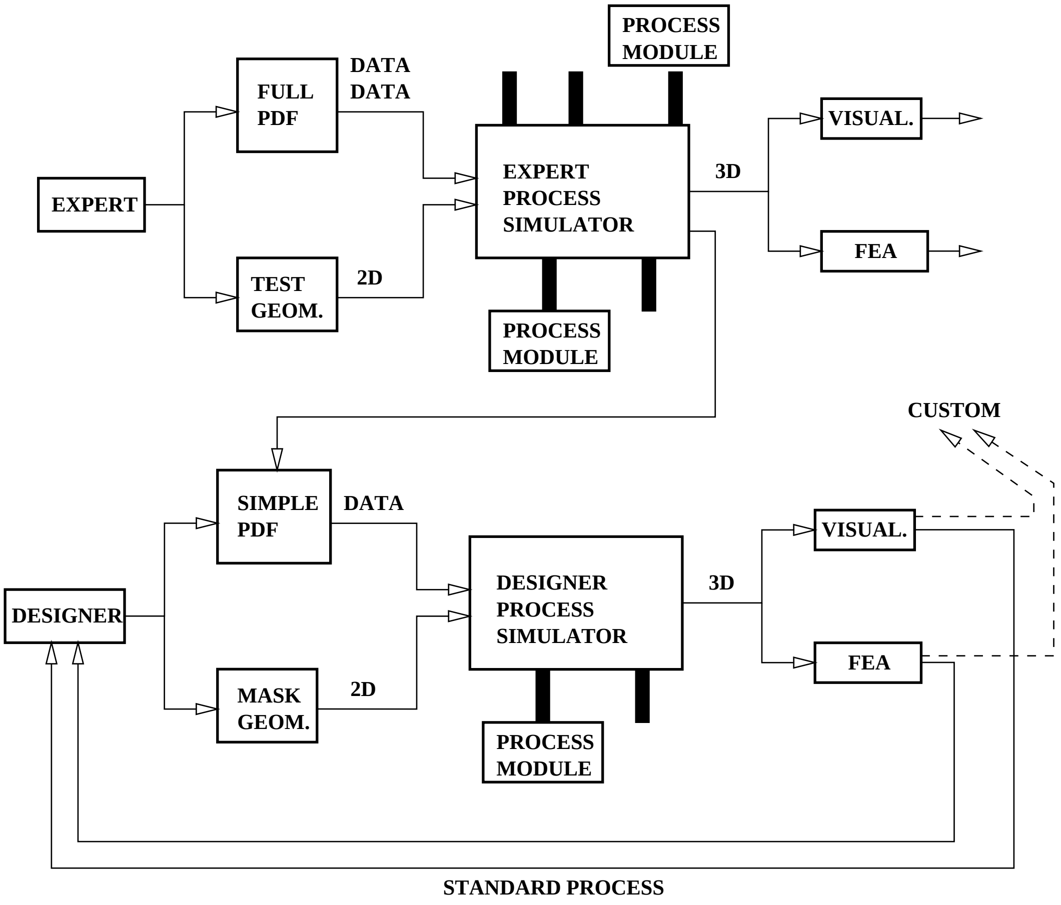


Figure 8: Simulation of Fabrication.

operates on PC's, and a high-end complete simulation package which is computationally intensive.

Figure 8 summarizes one potential approach. The expert designer who is doing custom or experimental design must be aware of the details of the fabrication. Thus he would need an expert process simulator. The process definition file (PDF) would then include a large number of process parameters. A novice designer would prefer a cleaner separation between design and fabrication, and would use a simpler designer process simulator, which would shield him from most process details. The associated PDF would be much simpler, perhaps generated by an expert designer to give a reasonable, simpler approximation to the process. In both cases there is a trade-off between process detail and clean separation. Different users will have different needs and it is important that the two tracks be maintained. The results of the process simulators are evaluated using 3-dimensional visualization tools and/or FEA tools. The designer may have to run the process simulator, examine the results, and modify his design a number of times in order to achieve an acceptable solution. If no acceptable solution is found by standardized systems, a custom or expert process may be needed.

Process Definition File

The process simulator constructs the 3-dimensional representation of a shape using a number of inputs. These inputs include the mask data and the PDF (process definition file) which contains a sequence of process steps that represent the fabrication process. The PDF must reference both layer information contained in the mask file and simulation code modules in the process simulator. Note that the PDF's for expert/custom process designers will have a much higher degree of detail and information than will PDF's for standardized process designers. The high level of detail in the expert/custom simulators will be used to design the process itself, and may contain proprietary information. Standardized PDF's will be an approximation designed to capture the essential result of the process, while separating the designer from the fabrication. Standardized PDF's may be created by the fabricator from a detailed knowledge of the process, or by the designer based on test fabrication and measurement. PDF's should be extensible since new processes will continue to be developed and a process which is custom today may become standardized next year. Specific investigations need to be carried out on the format of the PDF as well as methods to ensure a smooth interface between simulator modules and PDF's.

Continuous Time Sequence Evolutions

Compared to VLSI, MEMS has a greater need for 3-dimensional partial etch time snapshots and animation. This greater need is due in part to 3-dimensional etching (lateral/underetch and vertical etching). While there are many different MEMS fabrication techniques (surface, bulk, LIGA, deposition) their visualizations needs should be attainable by one set of tools. Some MEMS shapes change non-linearly in time, that is the interesting shape changes can occur in short periods of time: *e.g.*, compensating structures, intersecting shapes. Time evolution tools need to be developed and distributed to the MEMS community which will help answer the questions: When is the shape defined? What time portion is most critical to the definition of the shape?

Process Variability

No process is ideal and variation or noise will always exist. For example the etching process may depend critically on such factors as temperature, pressure, concentration and timing, all of which may have slight variations. Who has the burden of keeping track of process noise? The answer depends on the application: novice designers with standard processes will want to be shielded, while expert designers will want the full capability to simulate process variation. In VLSI, the design rules are conservative, so the user has little noise burden. MEMS may be closer to analog electronics where some burden remains with designers. What types of statistical tools are needed to model process variations? How is robust design to be incorporated into the process simulation, *i.e.*, how do we develop designs that are insensitive to process variations? For example in micro-electronics, an analog differential pair is designed so that the pair is relatively insensitive to process variations. How do we design “self-correcting” MEMS devices or “mechanical differential pairs”, that is mechanical devices that are insensitive to process variations. VLSI designers are trained to design robustly, MEMS needs the same design experience and techniques and such approaches should be explored.

Metrics and Performance Criteria

In order to evaluate the suitability of process simulators, it is necessary to measure the difference between the actual fabricated shape and the predicted shape. This will give a measure or metric of the fidelity of the simulator. What are the performance criteria, by which process simulators are judged? Standardized benchmarks or test suites should be developed that provide a means of evaluating simulator performance. Process simulators would then include a summary of benchmark results, these would be a record of past performance, not a guarantee or warranty of future performance. This is so because the performance criteria for simulators and the suitability of simulators will be application dependent. Thus the benchmarks should cover a wide range of application so that users may select the benchmark which best reflects their application. A set of benchmarks for MEMS process simulators should be developed.

3.4.2 Findings and Recommendations

Finding: Process Simulation Tools. Existing VLSI fabrication process simulation tools have been developed for electronic device fabrication, and may be appropriate for micro-mechanical structures.

Recommendations:

- Survey and formal study of state of the art of process simulators,
  evaluate existing VLSI and MEMS simulators,
  determine level to which existing tools can be utilized, [2-years]
- Incorporate versatile process simulators in MEMS,
  move from 1st order effects to 2nd and 3rd order effects,
  expand availability of tools to broad range of users, [5-years]
- Simulation engines are fully modularized,
  manufacturers provide plug-in virtual process modules. [10-years]

Finding: Type of Simulators. Different designers (and the design of different devices) will likely require different levels of interaction between design and fabrication.

Recommendation: Develop simulators of the fabrication process that permit (if necessary) the designer to adapt the design to the particular fabrication process, but also permit the designer to ignore the details of the fabrication process where possible.

Finding: Process Definition File. An extensible standard format for a PDF (process definition file) containing a sequence of process steps for the fabrication process is needed.

Recommendation: Initiate development of an extensible standard PDF format.

Finding: Continuous Time Sequence Evolutions. Simulation of MEMS fabrication will require continuous time sequence evolutions, rather than a single end-result.

Recommendation: Continuous time simulations need to be developed to help develop successful, robust, processes.

Finding: Process Variability. Process variability will always introduce variations and uncertainty into the manufacturing process.

Recommendation: Develop design and manufacturing processes that reduce sensitivity to these variations. Develop approaches to designing “self-correcting”, robust, MEMS devices.

Finding: Metrics and Performance Criteria. Accuracy of process simulators is difficult to establish, and may vary with process or device design.

Recommendation: Develop standardized benchmarks or test suites to provide a means of evaluating simulator performance.

3.4.3 Summary

There is a continuum of MEMS designers, which will require a range of process simulation tools. The two extremes of process simulations types are the library-based designs utilizing standardized foundry processes and the custom processes that involve the concurrent development of both the design and the process. We believe that a clean separation between design and fabrication can be attained in the first case, while clean separation is neither desirable nor attainable in the second case. The standardized processes lead to time and cost effective designs by taking advantage of economies of scale. The custom processes develop new processes, and custom processes of today are the standardized processes of tomorrow.

We recommend that the development of process simulation tools to support both tracks of designers be pursued. We feel that these tools will be an enabling technology in the realization of the full potential of MEMS.

3.5 Infrastructure

Finding: Teaching. One of the major recommendations of the 1994 NSF *New Paradigms for Manufacturing* workshop [1] was to create a national infrastructure for design and manufacturing of mechanical and electro-mechanical systems. In particular, it emphasized the creation of a technical community which will communicate and share design tools, fabrication processes, educational materials and technical expertise. Development of such infrastructure played a crucial role for the success of VLSI revolution. The idea of multiproject chips, the development of silicon brokerage service like MOSIS and the availability of public domain VLSI design tools linked to the fabrication facility provided a fertile ground for the university researchers and graduate students to conduct new experiments on computer architectures and contribute to the development of fundamental research on structured design methodology. The rapid spread of so-called “VLSI culture” was also enhanced by the introduction of VLSI design classes for faculty members, often conducted with NSF sponsorship, who brought the VLSI revolution in the classroom and into the curricula.

Recommendation: Drawing from the VLSI experience, this workshop recommends the creation of a national infrastructure for design and manufacture of MEMS. In particular, we recommend the creation of a set of standard MEMS processes in surface, bulk and LIGA micromachining, creation of a MEMSIS - a MEMS implementation service and development of courses and curriculum materials for MEMS design and fabrication. As part of this effort, we also recommend that courses be created specifically for faculty members and an annual Workshop with participation of industry and academia to stimulate future research and facilitate curriculum development in MEMS area.

4 Workshop Participant Position Papers

The following position papers were submitted prior to the Workshop by each participant. Each participant then had the opportunity to revise his or her position paper after the workshop, to be included in this report.

4.1 Structured Design Methods for MEMS

Prof. Erik K. Antonsson, Ph.D., P.E.
Engineering Design Research Laboratory
Division of Engineering and Applied Science
California Institute of Technology
Pasadena, CA 91125
`erik@design.caltech.edu`

While considerable progress has been made in the areas of etch simulation [4, 6, 7, 8, 18], finite element analysis [3, 5, 11, 14, 16, 17, 22], corner compensation [1, 13, 15], and design [10, 19, 20, 21] the fabrication of MEMS has been made without the benefit of design automation techniques. In contrast, the design of VLSI systems has become highly formalized and automated. One of the goals of Mead and Conway's early work in the VLSI area was to permit "ordinary engineers" to perform design [12]. Prior to their work, VLSI design was the exclusive domain of highly trained and experienced specialists. Other engineering domains (e.g., MEMS design) have not had the benefit of the same level of formalism and automation in design, and engineering design in these areas remains the province of highly trained and experienced specialists.[2]

Work recently begun in several research groups (including our own) aims to permit rapid, accurate, conservative mask-layout synthesis of MEMS in a way analogous to present-day VLSI design. The long-range objective is to enable a MEMS designer to specify a desired micro-mechanical function (e.g., a mechanical spring with particular characteristics), and have a system automatically generate the information (mask-layout, and other fabrication instructions) to create the shape that exhibits the desired function. This approach will mean that MEMS designers will be able to concentrate on the desired function of the device, rather than the details of its physical manifestation.

Thus the goal is:

- To develop a MEMS mask-layout synthesis methodology
  that will automatically create a mask-layout
  for a given desired final 3-D shape.

Attaining this goal will, however, be difficult. Mechanical systems are more complex than digital VLSI: there is no (single) language to describe mechanical function; the geometry of mechanical systems is typically complex (non-Manhattan); the number of primitive mechanical elements is large; there is no separation between function and form (the case of a mechanical transmission, for example, supports the bearings; contains the lubricant; provides the mounting; provides cooling; etc.). However, for suitably constrained mechanical domains (perhaps 2.5D multi-layer surface micromachining) some progress may be made.

[2] Here the term "formal" is used to mean computable, in the sense that a design process can be automated. There are many methods that are in daily use in mechanical engineering design, but are insufficiently formal to permit automation of the design process. "Systematized" is a commonly used synonym to "formal" as the term is used here.

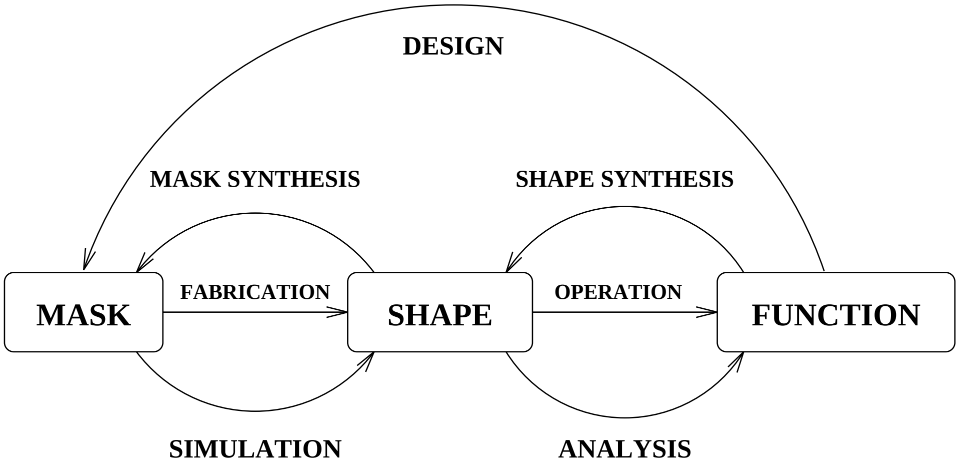


Figure 4.1.1: Simulation vs. Synthesis, Design vs. Analysis

It is important to observe that while design automation for digital VLSI has achieved considerable success, this is not the same a general electrical or electronic design. Many areas of analog or microwave (etc.) electronics design remains largely unautomated. Similarly, structured mechanical design methods will most certainly be developed for constrained (perhaps narrow) systems and applications.

The primary method for creating a MEMS mask-layout today is trial-and-error, guided by experience. Consequently, many iterations, and hence many prototypes, are typically required to develop a mask-layout that results in the desired shape and desired function. As Brysek, Petersen, and McCulley recently observed (1994):

> “In-depth knowledge of the [fabrication] process is needed because in micromachining ‘what you see’ is often NOT ‘what you get’.” [2, Page 25]

An illustration of this point is shown in Figure 4.1.2. This will be particularly true for future MEMS systems which will involve many degrees of freedom and/or complicated 3-dimensional shapes.

Because of the geometric complexity of surface fabricated MEMS devices, the present MEMS design procedure can be characterized as a *mask-to-shape-to-function* process. Even though the designer may start with a function and shape in mind, the complexity of the fabrication process forces the design cycle to iterate around the mask-to-shape-to-function evaluation process, as shown by the bottom arrows in Figure 4.1.1. However, the desired approach is exactly the reverse: *function-to-shape-to-mask*. That is, the designer conceives of a MEMS function, then through an automated (but perhaps iterative) process determines a shape that will exhibit the desired function, as shown by the top arrows in Figure 4.1.1. For example, the designer can develop a tentative shape, and then use FEA methods to

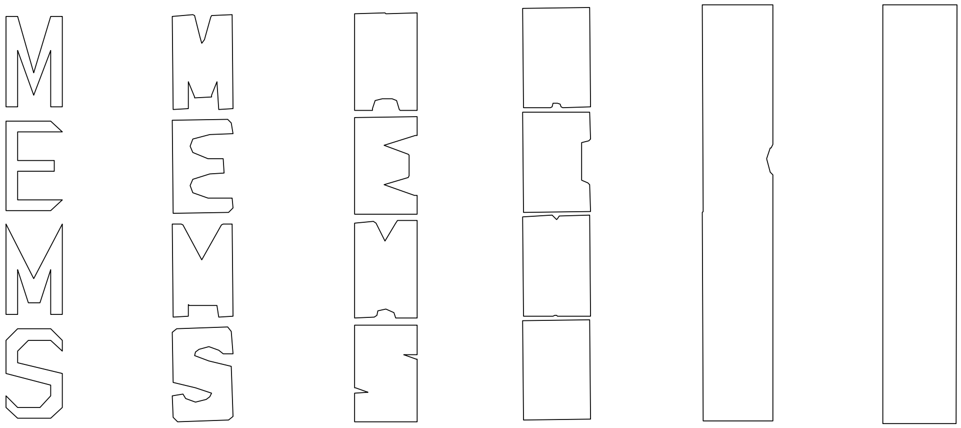


Figure 4.1.2: Input mask shape (on the left) and changing shape with time; anisotropic etchant simulation.

iteratively refine the shape until it exhibits the desired characteristics. Next, using an automated (but again perhaps iterative) process, the shape description is used to determine a mask-layout and a set of fabrication instructions that will create the desired shape, or the best possible approximation to that shape. In the case that the desired shape can not be fabricated, the designer may again need to use FEA tools to evaluate the suitability of the best approximate shape. For this reason, a standard communication format for the transfer of information between different levels of MEMS design is proposed.

Current FEA (and related MEMS CAD) approaches should be augmented to include the complexities of typical MEMS fabrication processes into the design cycle. Current MEMS FEA methods focus on the relationship between function and (3-D) shape. Additionally the relationship between (3-D) shape and mask-layout should be further formalized.

To develop formal and computable methods for the "shape-to-mask" process, a more exact computational model of the MEMS fabrication process is required. Such models will form the basis for the *forward* or *simulation* problem. That is, the solution to the forward problem determines what shape results from a given mask and a given etching process. More importantly, the model for the forward process is necessarily the basis for the *inverse* or *mask-layout synthesis* problem–i.e., determine the mask shape that yields a desired processed shape for a given etchant. One result that can commonly occur, is that no such mask shape exists. In this case, a *shape approximation metric* will need to be applied to determine the closest shape (or perhaps the closest function).

A closely related problem is that of the optimally "robust" shape. Even if a desired shape can be produced by an idealized fabrication process, deviations in the processed shape from the desired shape will occur due to process variations, small errors in mask alignment, errors in etch rate diagram data, non-ideal effects, and finite mask resolution.

One might alternatively define a *robustness metric* or a *sensitivity metric* for a shape. That is, how likely it is that the desired shape will be obtained (or obtained within an acceptable tolerance), assuming an expected range of processing errors? Analogously, how sensitive is a given shape to processing variations? Such metrics are useful for many applications. In the case that more than one mask shape will lead to the same processed shape under ideal conditions, the robustness metric could be used to select the most robust mask shape. The robustness metric can also be used to compare the output of different design procedures. Robust metrics may also be the basis for procedures that estimate process yields.

As the complexity of MEMS grows, the need for design automation will also grow. Design automation for MEMS represents a significant opportunity to build on the pioneering work in MEMS fabrication, modeling and analysis, along with the established work in VLSI design automation. The added complexity of 2- and 3-D mechanical devices introduce new challenges and will require considerable extensions beyond the VLSI domain. However, the inherent limitations (limited number of materials, limited shapes and sizes, limited forces, etc.) hold the promise for producing more tangible design automation results than have been obtained in the macro-mechanical domain.

References

[1] ABU-ZEID, M. Corner undercutting in anisotropically etched isolation contours. *J. Electrochem. Soc. 131* (1984), 2138–2142.

[2] BRYZEK, J., PETERSEN, K. E., AND MCCULLEY, W. Micromachines made of silicon. *IEEE Spectrum 31*, 5 (May 1994), 20–31.

[3] BUSER, R. A., CRARY, S. B., AND JUMA, O. S. Integration of the anisotropic-silicon-etching program ASEP within the CAEMEMS CAD/CAE framework. In *MEMS 92* (Institute of Electrical Engineers, 1992), pp. 133–138.

[4] BUSER, R. A., AND DE ROOIJ, N. F. ASEP: A CAD program for silicon anisotropic etching. *Sensors and Actuators 28* (1991), 71–78.

[5] CRARY, S. B., AND ZHANG, Y. CAEMEMS: An intergrated computer-aided engineering workbench for micro-electro mechanical systems. In *MEMS 90* (Institute of Electrical Engineers, 1990), pp. 113–115.

[6] DELAPIERRE, G. Anisotropic crystal etching: A simulation program. *Sensors and Actuators 31* (1992), 267–274.

[7] FRANK, F. C., AND IVES, M. B. Orientation-dependent dissolution of Germanium. *J. Appl. Phys. 31*, 11 (Nov. 1960), 1996–1999.

[8] FRUHAUF, J., TRAUTMAN, K., WITTIG, J., AND ZEILKE, D. A simulation tool for orientation dependent etching. *J. of Micromechanics and Microengineering 3* (1993), 113–115.

[9] HUBBARD, T. J., AND ANTONSSON, E. K. Emergent Faces in Crystal Etching. *Journal of Microelectomechanical Systems 3*, 1 (Mar. 1994), 19–28.

[10] KOTA, S., ANANTHASURESH, G. K., CRARY, S. B., AND WISE, K. D. Design and fabrication of microelectromechanical systems. *ASME Journal of Mechanical Design 116*, 4 (Dec. 1994), 1081–1088.

[11] MASEEH, F., HARRIS, R., AND SENTURIA, S. A CAD architecture for MEMS. In *Transducers 90* (Institute of Electrical Engineers, 1990), pp. 44–49.

[12] MEAD, C., AND CONWAY, L. *Introduction to VLSI Systems*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1980.

[13] OFFERIENS, H. L., KUHL, K., AND SANDMAIER, H. Methods for the fabrication of convex corners in anisotropic etching of (100) silicon in aqueous KOH. *Sensors and Actuators A25-27* (1991), 9–13.

[14] POURAHMADI, F., AND TWERDOK, J. Modeling micromachined sensors with finite elements. *Machine Design* (July 1990), 44–60.

[15] PUERS, B., AND SANSEN, W. Compensation structures for convex corner micromachining in silicon. *Sensors and Actuators A21-23* (1990), 1036–1041.

[16] SCHWARZENBACH, H. U., KORVINK, J. G., ROOS, M., SARTORIS, G., AND ANDERHEGGEN, E. A micro electro mechanical CAD extension for SESES. *J. of Micromechanics and Microengineering 3* (1993), 118–122.

[17] SENTURIA, S. D., HARRIS, R. M., JOHNSON, B. P., KIM, S., SHULMAN, M. A., AND WHITE, J. K. A computer-aided design system for microelectromechanical systems (MEMCAD). *Journal of Microelectomechanical Systems 1* (Mar. 1992), 3–13.

[18] SEQUIN, C. H. Computer simulation of anisotropic crystal etching. In *Transducers '91* (San Francisco, CA, USA, 1991), pp. 801–806.

[19] TRIMMER, W. S., AND GABRIEL, K. J. Design considerations for a practical electrostatic micro-motor. *Sensors and Actuators 11* (1987), 189–206.

[20] WISE, K. D. Integrated microelectromechanical systems: A perspective on MEMS in the 90s. In *MEMS '91* (Institute of Electrical Engineers, 1991), pp. 33–27.

[21] WISE, K. D., AND NAJAFI, K. Microfabrication techniques for integrated sensors and microsystems. *Science 25* (Nov. 1991), 1335–1342.

[22] ZHANG, Y., CRARY, S. B., AND WISE, K. D. Pressure sensor design and simulation using the CAEMEMS-D module. In *Tech. Dig. IEEE Solid-State Sensor and Actuator Workshop* (Institute of Electrical Engineers, 1990), pp. 32–35.

4.2 Structured Design Methods for MEMS

Dr. Selden B. Crary
Center for Integrated Sensors and Circuits
The University of Michigan
1126 EECS Building
1301 Beal Avenue
Ann Arbor, MI 48109-2122
crary@umich.edu

Summary

We believe that a software design system can now be envisioned for MEMS that will be able to take a set of desired performance requirements as input and accomplish the following tasks: choice of the optimal technology; selection and optimization of mechanical and electronic components from which the system will be composed; and determination of appropriate couplings among the components, including the degree of monolithic integration and packaging. The output will be a set of mask descriptions and an optimized process flow. Such a software system will require structured design methods, although these will, by the nature of the design problem, be largely distinct from the methods used in VLSI design.

1. Statement of the Problem

The study of microelectromechanical systems (MEMS) is a rapidly growing area of research with a large potential to accomplish useful tasks in numerous applications, such as the following: microsensors, microactuators, micro-accelerometers, microphones, cellular phones, and microelectromechanical filters. MEMS elements that have appeared to-date include rotary motors, linear motors and resonators, springs, gears, grippers, diaphragms, and arrays of mirrors for display technology. All of these elements and systems have mechanical structures on a size scale of a few to a few hundred microns. MEMS is a quintessential interdisciplinary field of electronics, bringing together studies in mechanical engineering, electrical engineering, electronics, fluid mechanics, optics, chemistry, and chemical engineering, with application areas across the entire spectrum of national and commercial enterprises. Partly because of its infancy, and partly because it involves such a large number of disciplines, there is not yet a developed science of design for MEMS. Teams of interdisciplinary researchers are needed with a common interest in establishing the required science and engineering for MEMS design, which we see as consisting of design synthesis and process planning. An important goal is the establishment of a set of methodologies for the design of MEMS that starts from a specification of desired function and leads to an optimized fabrication of a MEMS system.

2. Envisioned System

The system we envision will be able to choose the optimal technology, select and optimize mechanical and electrical components from which the system will be composed, and determine appropriate couplings among the components, including the degree of monolithic integration and packaging. The input to the design system will be a set of desired performance specifications, and the output will be a set of mask descriptions and a semiconductor-process flow. To attain the goal of a function-driven MEMS design system, there will of necessity be a need to develop parallel codes.

3. Library of Elements

A library of parameterized MEMS elements and functional building blocks will need to be established, including the interesting case of compliant mechanisms.

One existing approach to the construction of elements of a library of MEMS devices is that taken by CAEMEMS (Computer-Aided Engineering of MEMS). CAEMEMS is a framework for the design of MEMS that provides a high-level, Motif-based graphical user interface for the specification of specific instances of parameterized MEMS devices, generation of inputs to a finite- element analysis (FEA) system, launching of FEA runs on a serial workstation, retrieval of FEA results, stripping of results and storage in the database internal to CAEMEMS, capabilities for single-run analysis, multiple-run analysis, and sensitivity analysis, as well as plotting of families of line graphs of use for the designer.

The library of MEMS elements and blocks may take various forms, ranging from simple, e.g., the geometric and materials specifications of a simple beam element, to complex, e.g., a tensor spline model of a parameterized electrostatic motor, including possible couplings to other functional blocks. It may develop that the library will consist mainly of software that can construct models of elements and blocks, rather than simply consisting of the models of the elements themselves.

4. Compliant MEMS

Since the elements of the traditional mechanical repertoire consisting of rigid links and joints, fail to meet atypical requirements of micro-regime such as (i) eliminating the need for assembly, (ii) restricting the entire machine systems to just one or two layers in a plane, (iii) alleviating the adverse effects of friction, and (iv) accommodating unconventional actuation techniques including thermal, piezo-electric, etc. Fully compliant mechanisms, a certain kind of generalized flexible structures, readily meet all of these requirements.

Compliant mechanisms are single-piece flexible structures that deliver the desired motion by undergoing elastic deformation. This is in contrast to the rigid-body motions of conventional mechanisms. It will be important to develop mathematical formulations for optimal design of compliant micromechanisms. One of the computational procedures used in solving the synthesis problems is based on the homogenization method. This method, based on cellular microstructure, has the ability to generate any topology, shape, and size that are optimal for given problem specifications, which are applied forces, desired output displacements, and the amount of material to be distributed in a prescribed design domain. The output of this method is a density image in gray scale that indicates the optimal material distribution within the design domain. The homogenized image is directly transformed into manufacturable form. Structured methods are needed for the application of the homogenization method in the MEMS domain.

5. Process Synthesis

An important part of the proposed research involves the determination of process sequences for the fabrication of desired MEMS devices. This activity will be led by Prof. Carlos Mastrangelo. The design of fabrication processes for micromechanical and microelectronic devices requires a working knowledge of semiconductor thin film processing, along with a knowledge of materials and processes and a great deal of ingenuity. Some commercial VLSI processes require as many as 500-700 process steps, and a comparable degree of fabrication complexity occurs in the fabrication of three-dimensional micromechanical structures using planar processes.

In the micromechanics area, programs such as MEMCAD and IntelliFab give accurate representations of the finished micromechanical devices. These simulation tools take a description of the fabrication process flow and a mask set of a device as inputs and generate simulated profiles of the finished device as outputs. These design tools are undoubtedly useful aids to the designer that allow him to correct potentially expensive mistakes before fabrication begins. Nevertheless, these tools are aimed at design verification, and they require the input of, or assume, a known fabrication sequence, typically developed by experienced designers, hence the process flow is very much subject to their ingenuity and knowledge background. A more useful tool for the future a geometrical description of the device as input and provides, as output, a fabrication process flow. Such a program will speed up the development of microdevices by removing the knowledge background requirement from the human designer. The program will have the added advantage that it can be coupled to existing simulation tools to provide accurate information about the finished device in a true rapid concept-to-manufacturing design-synthesis fashion. Such programs must select the microdevice materials and fabrication sequence, as well as determine the feasibility of the decomposition of the given structure into layers. Furthermore, such programs will be useful in the design of robust fabrication processes, estimation of yields and manufacturing variations, as well as automated design centering. It is possible that such tools may create fabrication sequences that could not be conceived by the limited scope of human designers.

6. Parallel Computation

Of necessity, extensive use will be made of parallel computation, and many aspects of the parallelization of relevant software for MEMS design will be important.

7. Scientific Aspects

The development of a structured design methodology for MEMS, taken in the broad sense presented in this position paper, will require and enable an important set of areas of scientific research; including the design of computational experiments for response-surface generation over extended, possibly highly non-linear domains; methods for efficient and effective man/machine interaction; and algorithms and communication strategies involving highly parallel computation. One particular area that we wish to bring out here is the need for a unified theory of transduction that is coupled to design, and we close with two paragraphs on this important emerging area.

7.1 Unified Theory of Transduction A unified theory of transduction has begun to emerge from the research of Middlehoek, Ylilammi, van Duyn, Kirschner, and others. This theory identifies a set of signal domains (e.g., electrical, radiant, mechanical, thermal, chemical, and magnetic) each of which is characterized by its own set of variables, equilibrium constitutive equations, and non-equilibrium constitutive equation (e.g., the relation between flux of charge carriers and the gradient of electric potential is Ohm's law, which characterizes an energy dissipative process). There is unity in the theory because, under an assumption of local equilibrium, there exist relations among the variables across the signal domains, such as Gibbs relations, equations of state, and a balance equation for entropy.

The unified, irreversible thermodynamic theory of transduction can be generalized from a design perspective to enable the initial abstraction of a desired transduction function to its generic thermodynamic basis. From this abstraction various methods can be applied to enumerate and select a device or system realization of the transduction function. Implementation of the function, that is, the specification of the exact process steps and masks required, is the final step in this virtual prototyping procedure. We believe that software can now be developed to implement each step.

4.3 Structured Design Methodology for MEMS

Gary K. Fedder
Department of Electrical and Computer Engineering
and The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15203-3890
fedder@ece.cmu.edu

One important trend in microelectromechanical systems (MEMS) is toward monolithic systems where micromechanical devices are integrated with digital I/O, self-test, auto-calibration, digital compensation, and other signal processing functions. There is a growing demand in the MEMS community for rapid micromechanical design and analysis of complex systems involving multiple physical domains, including mechanical, electrostatic, magnetic, thermal, fluidic, and optical domains.

Pieces of the approach taken in VLSI design may be useful in developing structured design methods for MEMS. CAD for VLSI spans many levels of abstraction from materials, device, circuit, logic, register, to system level. At each of these levels, a design can be viewed in physical, structural (schematic), or behavioral form. A similar design hierarchy for MEMS is feasible and sorely needed. Analogous hierarchical levels up to the VLSI 'circuit' level are easily made; higher levels of abstraction may evolve for MEMS that are different from the VLSI paradigm. A first task in development of structured MEMS design tools is the formation of standard data representations and standard cell libraries. An enormous effort is necessary to identify and to model reusable MEMS processes, elements, devices, and architectures. MEMS CAD tools must be integrated, with appropriate links available to the designer to switch between different lateral views and hierarchical levels.

An initial wish-list in the MEMS CAD toolset includes:

- standard MEMS data representations and interchange formats
- standard MEMS cell libraries supporting behavioral, schematic, and physical views at all levels of abstraction (e.g., materials database, layout cells, schematic element library, and a system macro-model library)
- standard MEMS process-module libraries and standard process flows
- process simulation and visualization
- process synthesis and technology file extraction
- 3D rendering and animation
- 3D generation from layout and technology files
- layout of arbitrarily shaped objects with design rule checking

- layout synthesis and verification
- fast modeling and verification tools; coupled multi-domain, numerical analysis (e.g., finite-element method, boundary-element method)
- parasitic extraction to schematic and behavioral views
- macro-model parameter extraction from physical and schematic views
- multi-domain schematic capture (i.e., schematic view showing connectivity between mechanical, electromechanical, thermal, and circuit lumped-parameter elements)
- mixed-signal multi-level multi-domain simulation

Current MEMS CAD Tools

Several groups have existing research programs to address the deficiency in MEMS design tools. Examples from the U.S.A. include MEMCAD (M.I.T.) [1] and CAEMEMS (Univ. of Michigan) [2]; examples from Europe include CAPSIM (Catholic Univ. of Leuven, Belgium) [3], SENSOR (Fraunhofer Institute, Germany) [4], and SESES (ETH, Zürich) [5]. These tools involve general numerical analysis of layout and generation of macro-models for simulation. MEMCAD has evolved into a MEMS modeling framework with rapid self-consistent electromechanical 3D numerical simulation. Recent advances have been made in simplifying the input and visualization of 3D models of micromechanical structures from layout using the MEMBUILDER tool [6]. CAEMEMS is a framework in which the users chooses among modules that address specific design domains. CAEMEMS automatically generates a set of parameterized response surfaces by launching multiple finite-element analyses. IntelliCAD [7] available from IntelliSense Corp. is a commercial MEMS CAD tool with automated 3D modeling from layout and process integrated with numerical analysis. Other commercial tools by Tanner Research [8] cater to the MEMS community by allowing layout of non-Manhattan geometry and supplying MEMS technology files with design rule checking. These tools are definite improvements over use of Magic or KIC for layout and stand-alone numerical analysis tools (e.g., ABAQUS, ANSYS, Maxwell). More effort must be poured into fast multi-domain numerical analysis tools specifically tailored for MEMS design. MEMS process simulation and synthesis tools are needed and are being developed [9], but a discussion is outside the scope of this summary.

Current MEMS Design Practices

Current MEMS design practices focus on physical device and process development. A simplified design methodology is shown in Figure 4.3.1. Design concepts are implemented in a manual layout. The performance is then analyzed using numerical analysis tools, usually resulting in iterations on both the layout and the underlying process. The present state-of-the-art in MEMS CAD relies on device-level extraction of macro-models in a limited

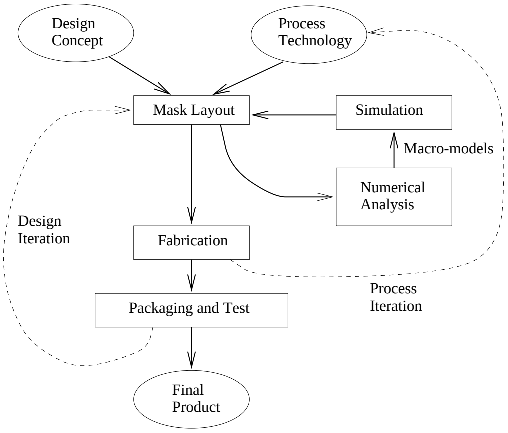


Figure 4.3.1:
Flowchart of current MEMS design.

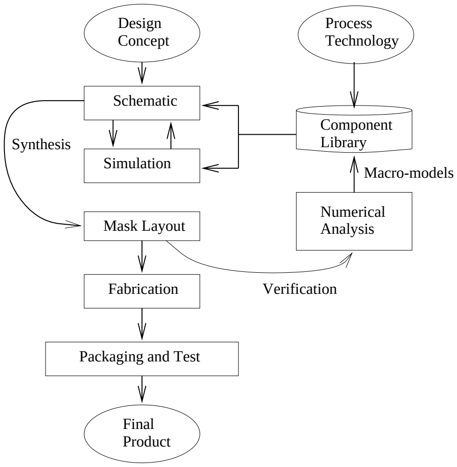


Figure 4.3.2:
Flowchart of structured MEMS design.

set of energy domains for behavioral simulation. Current commercial design tools cannot deal with the complex multi-domain architectures that will be necessary to create the next-generation of commercial MEMS. Much future work should focus on creating very fast multi-domain numerical simulation tools to ease both process development and device macro-modeling. However, these numerical tools by themselves may not be practical for rapid iterative design since the physical layout (and perhaps the process) must be changed for each iteration without necessarily knowing what to change to best to improve the device performance. Currently, a self-consistent electromechanical analysis of a simple device requires many person-hours to create the 3-D geometry and perform a numerical analysis. The manual design cycle in MEMS has not decreased significantly over the past few years since knowledge from previous development efforts cannot be easily reused by future developers.

MEMS Process Services

MEMS covers a broad, evolving spectrum of fabrication processes. This fact makes it very difficult to foresee the ultimate MEMS CAD framework. Our initial efforts at Carnegie Mellon have focussed on design tools for surface-micromachined MEMS, such as polysilicon MEMS built in MCNC's MUMPs process [10], and laminated oxide/aluminum MEMS built using MOSIS followed by an in-house dry-etch release step [11]. There are a several important benefits of making microstructures with stable foundry services such as MUMPs and MOSIS:

- sensor fabrication is fast and reliable,
- all, or most, fabrication steps are done externally, so research resources can be in-

vested in design, not standard processing,

- the process is repeatable, so circuit and microstructure designs can be re-used,
- devices improve as the process technology improves (e.g., scaling), and
- prototypes can be reproduced at any time.

Because of their planar ‘2 $1/2$-D’ topology, surface micromechanics is a MEMS technology which lends itself to abstraction in conventional schematic capture tools. Once a working structured design methodology is established for surface-micromachined MEMS, the techniques may be extended to other processes, such as bulk-machined Si or a dissolved-wafer process. The long-term goal is to enable rapid, intuitive exploration and analysis of the design space for complex MEMS.

Schematic Design and Synthesis of MEMS

At Carnegie Mellon, we are developing tools and model libraries to support schematic design and synthesis of MEMS, as shown in Figure 4.3.2. Physical and behavioral views of MEMS are currently used by designers, while schematic views have been neglected. The schematic provides a critical link between the physical and behavioral views. The designer is freed from doing detailed layout and 3D numerical simulation in the initial iterative design phase and can explore different design concepts quickly. In the MEMS schematic view, micromechanical devices are designed by assembling iconic representations of microelectromechanical lumped-parameter elements. Reusable elements, such as beam springs, plate masses, electrostatic actuators, and capacitive sensors, are backed by models of varying sophistication. Detailed models with second-order effects are used in simulations while simple first-order models are used in automatic optimal component sizing. The designer has freedom to experiment with new micromechanical architectures and then size elements appropriately. The key point is that new devices can be designed and high-level macro-models extracted without layout in the iterative design loop.

Efficient simulation is critical to the iterative design process. Mixed-signal mixed-domain simulation tools will be needed for the schematic design methodology. Many groups are already exploring behavioral MEMS simulation using device macro-models extracted from numerical analysis. Schematic design provides an accelerated method for generating higher-level macro-models, since they are constructed from schematic information and pre-made lumped-element models. Efficient 3D numerical analysis is critically important for generating lumped-element macro-models and for verification of final designs. I will re-emphasize that efforts must be made to link MEMS CAD tools together, including process simulation, numerical analysis, 3D rendering, layout, schematic design, and system simulation.

References

[1] S. D. Senturia, R. Harris, B. Johnson, S. Kim, K. Nabors, M. Shulman, and J. White, “A Computer-Aided Design System for Microelectromechanical Systems,” *Journal of Microelectromechanical Systems*, v.1, no.1, 1992, pp. 3-13.

[2] Y. Zhang, S. B. Crary, and K. D. Wise, “Pressure Sensor Design and Simulation Using the CAEMEMS-D Module,” *Technical Digest of the IEEE Solid-State Sensor and Actuator Workshop*, Hilton Head Is., SC, June 1990.

[3] B. Puers, E. Petersen, and W. Sansen, “CAD Tools in Mechanical Sensor Design,” *Sensors and Actuators A*, v.A17, 1989, pp. 423-429.

[4] B. Folkmer, H.-L. Offereins, H. Sandmaier, W. Lang, P. Groth, and R. Pressmar, “A Simulation Tool for Mechanical Sensor Design,” *Sensors and Actuators A*, v.A32, 1992, pp. 521-524.

[5] J. G. Korvink, *An Implementation of the Finite Element Method for Semiconductor Sensor Simulation*, Ph.D. Thesis, Swiss Federal Institute of Technology (ETH), Zürich, Switzerland, Nov., 1993.

[6] P. M. Osterberg and S. D. Senturia, “‘MEMBUILDER’: An Automated 3D Solid Model Construction Program for Microelectromechanical Structures,” *Technical Digest of the 8th Int. Conf. on Solid-State Sensors and Actuators (Transducers ‘95)*, Stockholm Sweden, June 1995, v.2, pp. 21-24.

[7] IntelliCAD, IntelliSense Corporation, 16 Upton Dr., Wilmington, MA 01887.

[8] L-Edit, Tanner Research, 180 North Vinedo Ave., Pasadena, CA 91107.

[9] B. Gogoi, R. Yeun, and C. H. Mastrangelo, “The Automatic Synthesis of Planar Fabrication Process Flows for Surface Micromachined Devices,” *Proceedings of the IEEE Micro Electro Mechanical Systems Workshop*, Oiso, Japan, Jan., 1994, pp. 153-157.

[10] Multi-User MEMS Process Service (MUMPs), MCNC MEMS Technology Application Center, 3021 Cornwallis Road, P.O. Box 12889, Research Triangle Park, NC, 27709-2889, WWW address: http://mems.mcnc.org

[11] G. K. Fedder, S. Santhanam, M. L. Reed, S. C. Eagle, D. F. Guillou, M. S.-C. Lu, and L. R. Carley, “Laminated High-Aspect-Ratio Microstructures In A Conventional CMOS Process,” *Proceedings of the IEEE Micro Electro Mechanical Systems Workshop*, San Diego, CA, Feb., 1996, pp. 13-18.

4.4 Getting a Clean Separation between Design and Fabrication

Jack Hilibrand and Bernard Chern
Division of Microelectronic Information processing Systems (MIPS)
Directorate for Computer and Information Science and Engineering (CISE)
National Science Foundation
Room 1155 4201 Wilson Blvd
Arlington, Va. 22230
`bchern@nsf.gov`
`jhilibra@note1.nsf.gov`

The heart of the VLSI design methodology, which has been so successful in the past two decades, is the clean separation between the designer and the fabricator. Carver Mead describes the critical choice of an interface to the fabrication community which would permit designers to focus on design and design methodology while not following (or trying to control) the details of the fabrication process:

> "The choice of the interface was made to coincide with the most generic definition of wafer fabrication: The final physical wafer would be built up of layers, applied one after the other. Each layer had a 2-dimensional definition. The design rules could be specified by simple geometric relations among the various layers. We specified the result of the operation on the wafer rather than how the result was to be obtained, thus leaving the processing detail to the particular fabricator."

The present practice in MEMS is for the designer to specify and send the mask set to the fabricator. The designer is necessarily involved in the details of the processing so that the mask set design will compensate for any and all divergences between the mask of a layer and the physical structure that is generated by that mask/process combination. A series of iterations enables the designer to bring the masks and the process in line to give the appropriate structures. Thus each mask set is good only for the specific fabricator since adjustments for process idiosyncrasies may not be appropriate for other vendors. If the designer describes the result he is seeking, the fabricators will order masks which make these adjustments and the design documents specifying the resulting object can be given to a variety of vendors for implementation, if desired. All will use different masks but deliver identical products.

It is hard to overstate the importance of this simple change (from describing the mask to describing the object) in the designer/fabricator interface in that it permits and encourages the desired clean separation. The clean separation results in and from fabricators identifying a set of standard processes for which they have detailed knowledge of the trims and adjustments needed in masking. It tends to defocus the designer from the process details to design concepts, design libraries, and design creativity where he can make his optimum contribution. Before VLSI, semiconductor designers did not observe this separation and the result was a highly fragmented design community where the best designs were the result of intense communication between the designer and the in-house model shop which built his parts. When the clean separation was in place, third party vendors were able to serve the design community effectively and design rules defined what could be done within the framework of available standard processes. These standard processes were updated and optimized by the fabricators to reflect progress in the process and equipment technologies and to keep them competitive within the fabrication community (independent of design community involvement). The basis of the clean separation is the existence and use of standard processes known to both the design and fabrication communities. This emphasizes the importance, for MEMS, of moving in the direction of standard processes where it is at all possible.

4.5 VLSI and MEMS, VLSI vs. MEMS

Ted J. Hubbard
Department of Mechanical Engineering
Technical University of Nova Scotia
Halifax, Nova Scotia
hubbardt@tuns.ca

Micro-electro-mechanical Systems (MEMS) is an emerging technology with great potential. Commonly known as micro machines, MEMS refers to mechanisms and circuits manufactured with feature sizes of several microns, allowing for large volume production and low unit cost.

Although the fabrication methods for MEMS are derived from the VLSI (very large scale integration) microelectronics industry, MEMS is an interdisciplinary field combining electrical engineering, mechanical engineering, materials science, physics and chemistry.

As MEMS moves from individual devices to systems with many interacting components, mechanical design will increase in importance. Macro or “real world” machine design has developed a large body of expertise in synthesising complex mechanisms. These methods and formalisms are essential tools in the growth of MEMS.

This position paper will examine VLSI and MEMS, VLSI vs. MEMS; that is both the similarities and the differences between the two.

There are many lessons to be learned from VLSI that are applicable to MEMS. VLSI is perhaps the most highly developed form of structured design. A designer is able to specify a high level description of a device without having to know the details of the fabrication. In fact this clean separation of design and fabrication means that a designer would prefer not to know the details of the fabrication process. The structured and separated design methodology of VLSI led to revolutionary reductions in design and prototyping costs. Can a similar advance be made in the field of MEMS, especially given that MEMS fabrication technology is derived from VLSI techniques? The question is particularly relevant since unlike VLSI, some macro mechanical research such as concurrent engineering seeks to remove the separation between design and fabrication.

There are specific differences between VLSI and MEMS that will require modifications to VLSI design methodologies for use in MEMS design. VLSI design is based on a set of conservative design rules, so that the designer can be confident that the process variations will not significantly alter the performance of fabricated devices. Designers sacrifice performance for design methodology. MEMS fabrication technologies are developed from VLSI fabrication technology, but they tend to “push the envelope” in order to produce desired mechanical properties. Thus there may be a smaller performance margin available for sacrifice. The more critically the operation depends on precise process specification, the less clean the separation between design and fabrication.

MEMS has traditionally been compared to digital VLSI (in this paper the term VLSI is used to mean digital VLSI), perhaps a closer match is analog VLSI. Like analog VLSI, MEMS has “piggy-backed” on VLSI technology. Analog VLSI design depends on a detailed knowledge of fabrication technology and there is a much less of a separation between design and fabrication. As in analog VLSI design, MEMS design sometimes involves breaking the VLSI design rules. Analog VLSI and MEMS rely much more on custom devices rather than the standard libraries of digital VLSI.

VLSI geometry tends to be Manhattan (rectangular). The macro mechanical world, shapes are inherently curvilinear, a multitude of shapes are needed. Rectangular shapes and sharp corners are usually avoided, if only to reduce stress concentrations. For macro mechanical devices form is intricately linked to function. The shape of a mechanism, (a cam for example) critically effects not only its efficiency, but its function itself. This stronger link will lessen the possible separation between design and fabrication. Related to form is the three dimensional nature of mechanical devices. The function of a mechanical device also depends on its three dimensional orientation and the orientation of its neighbors. The connections between devices is three dimensional, compared to the scalar behavior of electrical current. Thus the three dimensional routing of power (fluid, thermal, mechanical, and electrical) needs to be considered.

VLSI devices can be manufactured from a relatively small set of primitives, a small library can be quite rich in content. Macro mechanical devices have a much larger set of required primitives; the minimum size of library for rich content may be quite large. The move from custom designs to standardized libraries may be delayed by incomplete libraries. MEMS devices which have large numbers of devices tend to be arrays of identical arrays, i.e., mirror arrays. This approach is closer to VLSI with its memory arrays, than to macro mechanical systems where many different devices are interconnected in one system. There may be two causes: small libraries and difficulty interconnecting different parts.

In summary, MEMS is often seen as bridging the gap between VLSI and macro mechanics. It offers an opportunity to apply VLSI structured design methods to mechanical systems. However, it is important to stop considering MEMS as peripherals to electronics. Their fundamental mechanical nature will necessitate a reformulation of existing VLSI design methodologies. VLSI CAD tools cannot simply be repackaged and used for MEMS, the experience of the macro mechanical research community should be incorporated. Having said that, VLSI structured design tools represent the foundation upon which MEMS structured design should be built. With a firm design foundation, MEMS can realize the same growth as VLSI.

4.6 Structured Design Methods for MEMS

Ramaswamy Mahadevan
MCNC, MEMS Technology Applications Center
PO Box 12889
Research Triangle Park, NC 27709-2889
`ramu@mcnc.org`
`http://www.mcnc.org/HTML/ETD/MEMS/memshome.html`

Is there a sufficient similarity between MEMS and VLSI systems to borrow from VLSI design techniques?

VLSI systems use a small number of elements (devices) - transistors, capacitors and resistors - to build all other functional blocks and systems. Unfortunately MEMS does not appear to have as small a set of elements that can be utilized to synthesize any generic system functionality with a similar degree of flexibility. It should be possible to show that a moderately large functional design space could be covered using multiple instances and interconnection of elements from a small set of basic electromechanical elements, structured design techniques would be an immediate by product. (Perhaps along the lines of synthesis of mechanisms using four bar linkages?) Currently, MEMS design appears more akin to analog VLSI design; functional blocks like actuators, sensors, mechanical suspensions/springs, bearings, etc. are used in lieu of operational amplifiers, capacitors, resistors, and switches; interaction between blocks has to be taken into account.

Unlike VLSI, in MEMS the 2-D planar topology alone is not sufficient to model the electro-mechanical behavior of the system. It is necessary to incorporate process cross sectional information as well to generate the 3-D structures that will result when using a given mask layout - process combination. This requires the use of suitable device and/or technology models or representations. A few CAD programs do take mask and process descriptions to generate the 3-D structure that would result. However, there is a need to standardize the process description and database formats. There is also a definite need for layout representations and mask pattern generation software that support arcs and freeform structures in an efficient and accurate manner. (This also impacts on the time and cost of generating a mask that contains many arcs or freeform polygons.)

Unlike VLSI, separation of form and function may not be present. For surface micromachined devices, function of a device may be considered to be relatively independent of its position and orientation. However, the relative positions and orientations of devices in a bulk micromachining process do determine the structural shapes and functional behavior of the device.

Like VLSI, MEMS wafer fabrication and design can be considered to be cleanly separated where surface micromachined process are concerned. In fact independence of the fabrication process from the mask pattern is generally the desired process ideal sought. However, in some MEMS processes incorporating electroplating (such as LIGA), the fabricated device structure can depend on the mask pattern and/or fill factor of the mask pattern. (In addition, in MEMS processes such as bulk micromachining the fabricated structure also depends on the orientation and size of the layout pattern; however this can be accounted for in a process model.)

Despite these differences, MEMS can borrow from many of the VLSI design methods and CAD tools. Top-down and bottom-up design, synthesis, device and/or block level detailed simulation, functional or behavioral models and system level simulations are all equally applicable to MEMS. Even a version of LVS is possible for MEMS. The main obstacle hindering the easy use of such a approach is the multiplicity of physical phenomena that may be utilized in a MEMS system requiring (possibly coupled) mechanical, electromagnetic, fluid, and thermal analysis. For detailed simulations (equivalent to SPICE level simulations in VLSI) this can be done using existing FEA or BEA solvers. This requires an open CAD framework with a database format that includes mask layout and process technology information (process cross section and related mechanical properties) that can be easily exported/imported to/from the appropriate existing FEA solver. The alternate approach would be to use an equation based solver to model all the phenomena. SPICE itself can be used for functional or behavioral simulations of MEMS at the system level. There is also a need for simulators that are faster than FEA (and perhaps less accurate) analogous to switch level simulators in VLSI design.

The implementation of the following would help facilitate structured design and virtual prototyping for MEMS.

(a) A layout and process representation that allows efficient representation of arcs and freeform shapes along with process crossectional information, material properties, and perhaps even mesh information.

(b) An integral layout and solid modeling/viewing tool to help designers visualize 3-D structures that will result from the combination of layout and process description. (Existing tools can be adapted.) Visualization of 3-D structures that result from a given layout and process appears to be the biggest obstacle for new MEMS designers.

(c) An open CAD framework and data translators for sharing design information between existing and new MEMS specific CAD tools.

(d) MEMS specific mesh generation and analysis tools that are optimized for structures with thin layers (i.e., where one dimension is much smaller than the others).

(e) Improved simulators for computationally efficient modeling of MEMS at block and system levels.

(f) A CAD tool for the synthesis of mechanisms and/or structural elements.

There are various existing CAD tools that already have some of these features. For example, Stanford's TCAD VIP-3D, MIT's MemBuilder and MEMCAD, and IntelliSense's IntelliCad have layout and process description, 3D model extraction and mesh generation features. IntelliCad does have an integrated framework including process description and modeling, layout, solid model generation, and FEA analysis.

4.7 IntelliCAD: The CAD-for-MEMS

Fariborz Maseeh
IntelliSense Corporation
16 Upton Drive
Wilmington, MA 01887
IntelliS@aol.com

Introduction

U.S. industry faces a large opportunity with microsensors and other microelectromechanical (MEM) devices. Because these devices are smaller in size and can be made at a very low cost, they are not only replacing traditional sensors but can find applications in new areas. The automotive industry, in particular, has invested heavily in MEM sensors because of the tremendous cost savings. Five years ago the annual production of MEM sensors for engine and transmission controls and diagnostics was more than 12 million units per year in the U.S. [*High Technology Business,* Sept.-Oct. 1989, p. 29]. Indications are that this number has steadily increased since then with the drive for ever better mileage and lower emissions. Another large automotive application is silicon accelerometers for air bag release, anti-lock brakes and smart suspension. In 1994 the market for micromachined sensors was approximately $1.4 billion with applications in automotive, biomedical, process control, aerospace among others.

Silicon micromachining promises to dramatically reduce the cost of MEM devices because the process technology is very similar to that used in the high-volume manufacturing of semiconductor integrated-circuits (ICs). Rather than building electrical components like transistors and capacitors in silicon, manufacturers micromachine electromechanical elements for sensors and actuators. These miniature devices, with geometries measured in microns like ubiquitous ICs, have potential applications wherever traditional devices are found.

With the advantages of lower cost and widespread knowledge of semiconductor process technology, why is the MEM device market so small today? The reason is the lack of manufacturing knowledge in making the tremendous variety of miniature mechanical components. Semiconductor ICs have the advantage of commonality: the technology advances made for high-volume DRAM production spread quickly to other devices such as microprocessors and other types of memories.

The proper design and fabrication tools do not exist for MEM devices. Today's computer-aided design (CAD) systems are inappropriate because they do not have the necessary MEM design, material or fabrication information. As a result, manufacturers are forced to use time consuming and expensive design-fabrication trial and error to develop MEM devices. This enormous expense can be justified today only in applications such as in the automotive field which can amortize the costs over large volumes.

To address the need for design tools in MEMS, IntelliSense has developed an integrated software system, called IntelliCAD, in which a number of tools are integrated to assist developers of MEMS mimic the fabrication and modeling of MEMS in a workstation environment.

Users can design new micro fabrication processes with IntelliCAD by grouping logical sets of process steps into a process or by borrowing process steps from existing processes. Once a process exists, a user can change the physical makeup and material properties associated with that process by varying individual process components. Changes in the process components can then alter the corresponding look and performance of the device. The user can view a three dimensional representation of the resulting structure and can test the performance of the structure on-line by applying simulated loads and reviewing calculated effects. By using IntelliCAD, the entire design and manufacturing of a micro structure can be performed without entering a micro fabrication facility.

IntelliCAD Concept

A number of tools are integrated into a software system in which a user investigates a device from the conceptual stage to manufacturing. Each tool provides a specific functionality and is a transparent constituent of IntelliCAD. During the operation of IntelliCAD, the user interacts with the User Interface to perform various functions and receive information in a menu-driven or graphical format.

A process is first conceptualized through the Process Construction module with the Fabrication Database, and verified with the Process Check and Fabrication Simulation modules. By going back and forth among these modules the user will construct the process steps for the fabrication of a device. The process steps include a complete operational recipe with all the parameters required for an operator to perform each step. The user will periodically consult with the Fabrication Database and Material Database to select the parameter values for obtaining the material behavior of choice. The Material Database of the system has already been developed by IntelliSense to the point of commercialization. This tool, MEMaterial, consists of a material database, an estimation routine to simulate the material behavior of deposited films as a function of their manufacturing and a graphics user interface. Tool Interface passes material information to other constituents of IntelliCAD when needed.

The Solid Modeler, upon receiving the fabrication and material information from the aforementioned modules and the planar geometry information from the Layout Definition module, builds a three-dimensional model which can be used by the performance simulators for the analysis of mechanical, electrical, thermal and flow properties. The 3D models can also be displayed within the Graphics UI for visualization purposes at each step of the fabrication process. The Layout Definition module holds the mask information needed for definition of patterns in a process step. The layout information will be captured during the process construction.

Often users need to access standard (baseline) processes leading to the development of a certain MEM device. Standard process techniques are captured within the Design Database which can be accessed independently, modified when variations are needed at several steps. Simulations determine the effects of variations. The template processes stored within the

Design Database may be tailored to capture an organization's standard or proprietary process. The Design Database contains information about sensors (products), both a sensor's performance and its design.

IntelliCAD capabilities

Design Capabilities A microstructure design process is made up of a group of fabrication steps (e.g., deposition, etching, diffusion). A user needs to create a complete list of fabrication steps to define a process. To do this, a user can either customize an existing process or define an entire set of process steps based on a concept. Often a design solution will incorporate both of these techniques. To support this process, IntelliCAD includes a database of complete development processes as well as detailed information of individual fabrication steps. This information gives a user process design freedom while also providing defined processes that can be used as design starting points to save time.

IntelliCAD includes simulation tools that define new fabrication steps and verify the integrity of design processes. The fabrication simulation tool is used for creating variations of existing fabrication steps and can model the majority of known fabrication process steps. To perform this simulation, a known algorithm that defines physical design characteristics is invoked on a given process step using the characteristics of the process as input parameters. For fabrication steps that do not have known algorithms, a user defined algorithm can be user entered to override the fabrication simulation routine. The other simulator is the process simulation tool which is rule-based and contains fabrication process sequence rules developed using the collection and organization of micro fabrication expertise. It is used to verify process sequence continuity and completeness. If a process is not presented logically, the process simulation tool will report logic problems for user investigation. Before three dimensional visualization and performance simulation can occur the process simulation tool must verify process (or partial process) integrity.

Layout Definition One step in designing a process sequence is defining a mask or sequence of masks that will create an anticipated geometry. In IntelliCAD a user can choose a two dimensional mask and can view the geometric impact that the application of that mask (with given etch parameters) would have on a structure. A user can experiment with different masks and masking sequences in an effort to create a desired geometry. The layout definition feature replaces the mask definition, preparation, and application procedures that occur in the micro fabrication facility.

Material Properties Definition For deposition process steps, the designer has the ability to manipulate deposition thickness using the fabrication simulator and has the added capability to manipulate the material properties associated with thin film depositions. IntelliSense has commercialized a material analysis tool called MEMaterial that examines the material properties of thin films as a function of process parameters. The user can change parameters and MEMaterial will predict resulting material property changes. IntelliSense has integrated the MEMaterial product into the IntelliCAD package. When a deposition step is defined in IntelliCAD, the user can access MEMaterial and model the material properties of the deposited thin film. Thus the user can control the physical properties and material characteristics of thin film depositions.

MEMaterial contains a database of microelectronic materials that can be analyzed using estimation and optimization routines developed at IntelliSense. These routines compute material properties (based on measured data) for any given set of parameters. MEMaterial also allows users to analyze data using two and three dimensional graphs and generates plots that give users insight into material parameter dependencies.

On-line Visualization IntelliCAD contains a Solid Modeler Tool to provide on-line three dimensional visualization of user defined processes. As input, the solid modeler uses out-of-plane geometry from the fabrication and process building routines and in-plane geometry from the mask simulator. Based on the geometries inherent in the structure the Solid Modeler creates an on-line representation that appears as if it were fabricated in the lab. As an additional feature, the user can change the viewing perspective of the three dimensional structure to simulate a lab technician rotating the device to enhance viewing. The user does not need to create a complete process to use the solid modeler. Any group of coherent process steps can be evaluated.

Process Simulation The final step of the micro structure development process is performance simulation. The geometry that results from the Solid Modeler combined with the associated material properties from MEMaterial are used as input into the process simulation module. Performance simulation is achieved through finite element method (FEM) analysis. Multiple performance simulators including thermal, mechanical, electrostatic and electromagnetic types are interwoven within the IntelliCAD performance simulation module to create compete analysis and testing capabilities. From the GUI the user can apply various loads to a structure and view the effects that the loads have on the structure.

Graphical User Interface The graphical user interface is completely menu driven and all functions are clearly defined. A new user should be able to use the IntelliCAD software without referencing the user manual.

Data Availability The scope and magnitude of micro fabrication data associated with micro fabrication techniques and processes is continually expanding and it too fragmented to be captured in a single data repository. Therefore, IntelliCAD databases will provide a wealth of verified micro fabrication data and will allow users to incorporate their own data into a user defined data area. The user has the option of performing analysis based on IntelliCAD data only, user defined data only, or a combination of user defined data and IntelliCAD data. In addition to creating facilities for users to enter their own process data, IntelliSense is committed to providing updates of all IntelliCAD databases on a regular basis.

Conclusion

Microelectromechanical system (MEMS) technology emerged as an off-shoot of the semiconductor industry and much of MEMS technology is borrowed from the semiconductor industry. However, MEMS technology has unique manufacturing problems that require solutions that can not be borrowed from the semiconductor industry. A large demand exists for MEMS technology but we have failed to adequately manufacture the supply. Fast access to high quality manufacturing know how is essential for the MEMS market to reach its potential. The aim of this presentation is to target the obstacles responsible for the slow growth of this high potential technology and to suggest ways of overcoming them.

IntelliCAD is aimed to circumvent some of the difficulties present in the MEMS manufacturing. It promises to reduce the number of design-fabricate-test iterations required to manufacture a MEMS device. The reduced number of iterations will translate to reductions in time to market and manufacturing cost and will act as a catalyst for new and creative micro fabrication efforts. Currently, there are no fully integrated micro fabrication tools that support the manufacture of MEMS structures.

IntelliCAD will provide the following benefits to the micro fabrication community:

- shorter product development cycle times,
- shorter cycle times will create more demand from industry,
- development costs will be reduced,
- creativity and ingenuity will be inspired by faster time to market and reduced costs,
- information access: as the tool matures the data repository will become comprehensive.

4.8 Structured Design Methods for MEMS: Essential Tools

C. H. Mastrangelo
Center for Integrated Sensors and Circuits
Department of Electrical Engineering and Computer Science
University of Michigan, Ann Arbor
Room 1243
EECS Bldg.
Ann Arbor, MI 48109-2122, USA
carlosm@eecs.umich.edu

Introduction

Currently there are few or no structured design methods for MEMS at any level of abstraction. These levels include (a) MEM device simulation, (b) functional modeling for system simulation, (c) physical representation formats, and (d) process description specification. Historically, CAD tools for MEMS design have not been implemented until recently [1, 2, 3, 4, 5] on just a few of these areas. Therefore many MEMS designs have been developed by hand calculations, with at best a combination of existent tools developed for other older fields. In order to understand the MEMS designer task, and the possible identification of parts that can be aided by structured methods, it is instructive to describe a typical design cycle (Figure 4.8.1).

The design cycle for a MEMS device consists of several phases. The first phase involves the identification of the structural element needed (this itself is an *ad-hoc* process) followed by hand calculation analysis and first first order estimation of dimensions. In the second phase, the designer must create a realistic process flow for the desired device. He may be able to fit his design in a foundry-available process but, in general, depending on the nature and complexity of his design, the designer must resort to a custom process. During the process construction, the designer must be familiar with processing details such as material incompatibilities, layer etchants, and good, stable, and stress free thin-film materials appropriate for his design. This is undoubtedly a task that requires a great deal of process knowledge, and it is a major time consuming step of the design cycle phase.

Along with the process, a mask set must be generated to form the desired device under the process. This is most frequently accomplished using a conventional layout editor for VLSI. There are typically no guidelines available regarding design rules of any kind, with the exception of a few rules of thumb used to avoid problems related to the undesirable effects of undercut and stringers. In this stage it is typical to borrow some of the process CAD tools developed for semiconductor processing (such as SUPREM) which help determine the shape assumed by diffused and reactively grown layers. These, however do not provide much information regarding stress effects which ultimately may lead to device failure.

In the third phase, a good designer will construct a solid model of the actual micromachined structure and run a simulation on a conventional FEM system such as ANSYS. If the behavior of the device is somewhat satisfactory, he can always “optimize” its performance by adjusting some of the physical parameters such as mask features or thicknesses adjustments. Much of the design work discussed so far is trial-and-error. That is, if the simulations do not behave properly, the whole process must be restarted!

In passive MEMS devices, this may represent the end of the design cycle. However, in the construction of systems, the thin film structure must interact with electrical signals in a predetermined fashion. For example in a capacitive pressure sensor, the diaphragm deformation must be converted to a capacitance change. Similarly, actuators driven by electrical forces must have a characteristic transfer function. Whenever the device is connected to a circuit and feedback is present, the dynamics of the MEM device are interlinked with that of the circuit which, unless properly accounted for, can yield incorrect signal levels or, to an extreme, oscillatory behavior. To date whenever this situation is encountered, MEMS designers resort to simplified mechanical models (typically one or two pole approximations) of the distributed device dynamics. The link between the mechanical and electrical world is established through equivalent electrical models representing the mechanical variables of the system. This is in general possible because of the generality of Kirchoff’s laws.

For example, temperature and pressure can be represented by voltages while heat flux and mass flow rate can be represented by branch currents. In a practical implementation these electrical equivalents are implemented within a circuit simulator such as SPICE or SABER. The construction of distributed models for these is also possible through suitable discretization and generation of equivalent network, but this process is at best tedious. From the discussion above, it is evident that the design cycle involves many steps. Therefore structured design has the most impact only if a comprehensive design system or tool set is available. It is important to develop a system that can provide automatic design support at all stages of the cycle leading to rapid prototyping of MEMS.

Applications of Structured Methods

In order to facilitate the designer task many of these phases can be highly structured. In the simplest form, structured methods may develop in the form of standardized representation languages, such as CIF, GDS, and other layout languages are to VLSI devices. The formal definition of these representations must account not only for the mask layout but also for the depth of its layers. This is necessary as in general the device process will not be universal, and the individual layer thicknesses are crucial to the mechanical properties of the device. Furthermore such representation is very useful to perform design rule checks and to pinpoint potential process problems associated with the layer order and thicknesses.

At a higher level lay device simulation tools. These tools must be able to couple interactions between variables of different types. For example in actuators, the simulator must solve electromagnetic field coupled with mechanical deformation of the electrodes simultaneously [1, 3]. While accounting for all possible interactions across a large number of signal domains seems a formidable task, recent efforts in Europe suggest that a unified field theory of interactions can be achieved through generalized thermodynamical state equations [7]. This approach seems quite attractive since all properties of a domain are the result of energy exchanges and transport effects. Another approach currently assumed is the solution of independent decoupled problems with self-consistent boundary conditions.

Micromechanical process simulation tools are needed to model not only the physical and electrical behavior of thin films during processing but also the mechanical properties of the films themselves. For example, many of the layers used in these devices contain substantial levels of stress. This stress and the ultimate mechanical properties are functions of specific deposition conditions and subsequent thermal history. It is extremely useful to keep track of the "mechanical state" of the layers during the process in order to assure that the device layers will maintain mechanical integrity resulting in proper device operation.

Once appropriate numerical models for the device behavior are found, it is necessary to devise simplified component macromodel behavioral models to be coupled with systems. For example, portable modular component models generated from the simulation analysis are very useful if they are easily incorporated into circuit simulators. In fact, this is extremely important to accurately model the microstructure dynamics when connected to a circuit, and it is an essential tool needed for the implementation and simulation of systems. Recent activities in Europe are leading to the establishment of a hardware description language (VHDL-A) for MEMS similar to the well known VHDL language used in logic chips.

In order to perform all of these functions, each of these tools must be connected easily to each other. This can be accomplished through a series of translation tools between several standard description languages. For example, SIF may be used for physical layout while VHDL-A for behavioral, etc.

Top Down and Bottom Up Approach

While all of these tools are undoubtedly useful for speeding up a MEMS design cycle, the ultimate goal of the designer is to develop a MEMS structure that performs a specific function meeting desired specifications. The tools described above work on an "open loop" or "bottom up" mode. That is if the device does not meet the design criteria, the designer must make appropriate changes and repeat the cycle. If the device is complex, this process requires many iterations. The bottom-up approach thus demands an intense number of simulations per device developed, and it is indeed quite tedious.

It is highly desirable to develop a tool the automatically design the device starting from its specification. Compilation or "top down" tools do just that. In VLSI technology silicon compilers can design entire chips. In MEMS, the requirements of such system are different, and this function may be performed at different stages in the design cycle in order to avoid tedious manual iterations (Figure 4.8.2).

To date, it is unclear how a particular structure must look like to perform a given function, and as a result, the choice of device topology is dependent on the designer skill. In order to minimize the device complexity, it is very desirable to develop theoretical methods that essentially determine the number of device layers required to perform a given function. To date these methods do not exist, but research efforts are underway using a generalization of ideas borrowed from the well developed theory of mechanisms.

One of the areas that benefits the most from the top-down approach is that of process design. Since many MEMS devices require custom processes, their design inherently requires a broad spectrum of micro fabrication processing knowledge. This knowledge requires years of experience to acquire; therefore it is very desirable to develop a software tool that directly creates a process flow starting from a geometrical description of the device. Such tool has been already developed [4]. The methods utilized rely on a mathematical representation of the sequential fabrication process and the final device structure. The key element in the generation of a process flow resides on the order between the constitutive device layers. The device layer order can be represented in most cases by a directed acyclic graph in the form of an adjacency matrix. Further processing constraints can be easily incorporated as additional restrictions in the matrix; thus representing a compact means for the specification of good process design practices. Choices of etchants and deposition recipes are chosen from an internal database of lab-dependent and general data. The process flow is essentially constructed as an expansion of a topological sort of the graph nodes. Such procedure has satisfactorily assembled entire process flows for merged micromachining-MOS devices as well as complex 20-mask BiCMOS processes. Along with process compilation tools comes the development of a process language. This is in fact an essential need for communication with a foundry service facility, and the possibility of starting runs remotely on demand at the foundry.

Dimensional optimization [2] also plays an important role in the top-down approach. The behavior of distributed electromechanical elements can be tuned by proper adjustments of its dimensions. This can be performed automatically in order to meet the desired performance specifications. There are currently two approaches in this area. In the first, the behavior of each MEM unit element can be parameterized and stored as part of a library. Actual dimensional parameters can be determined through numerical optimization of nonlinear constitutive relations. In the second approach, the actual shape of the element can be determined to meet specifications. This approach, known as homogenization [6], is very attractive since it can yield electromechanical systems with simpler process flows and therefore a lower cost.

Summary

There are few or no existent structured design methods and CAD tools for helping researchers develop MEMS devices. The development of standard device representations, cross-field simulators, material property process simulators, and behavioral macromodels can shorten the design cycle substantially. The development of top-down compilation tools is the ultimate goal resulting in a “device-on-demand” technology with extremely short design cycles. While the development of any of these will shorten the design cycle for MEMS, their overall impact can be accomplished through the realization of a complete design system with few or no intermediate manual steps.

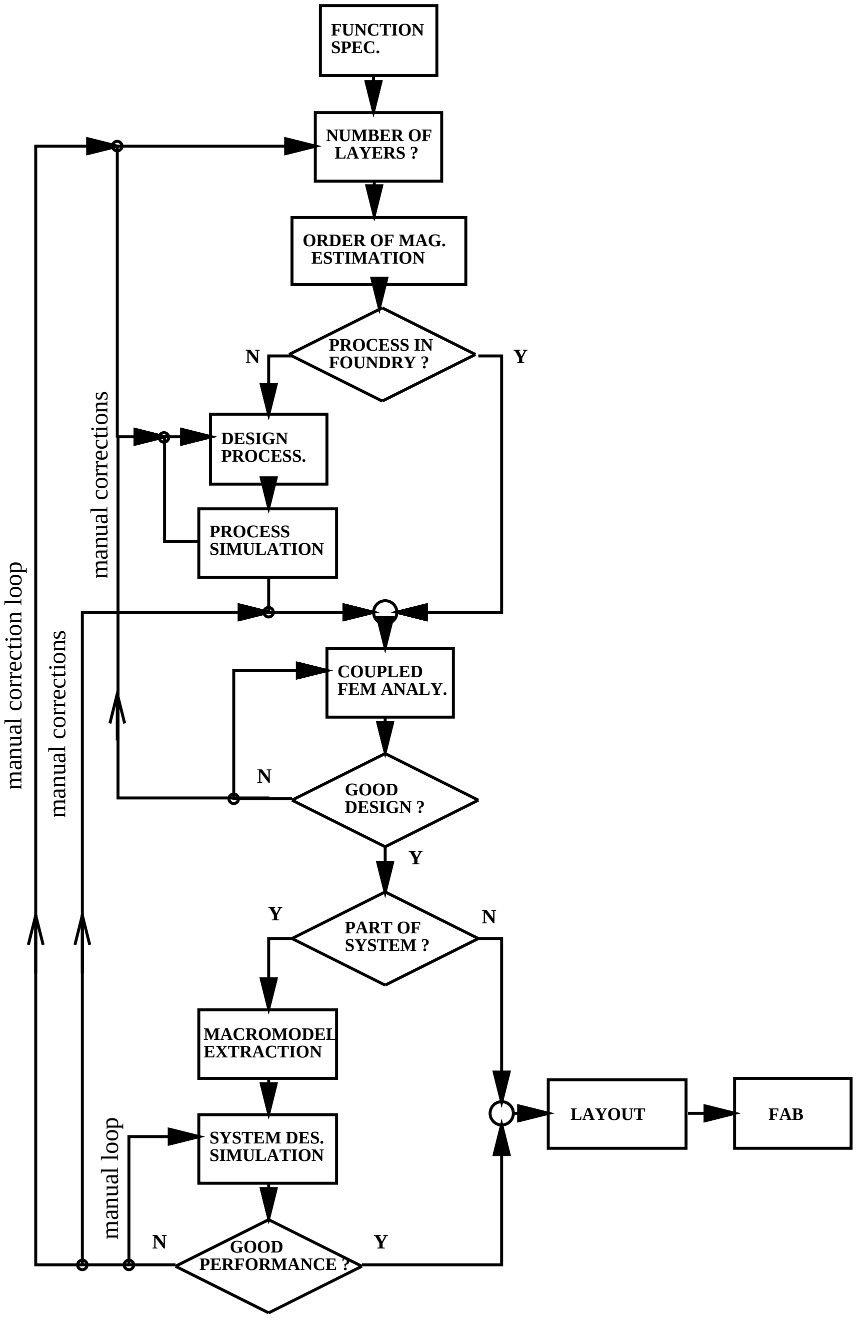


Figure 4.8.1: Typical MEMS design cycle (bottom-up)

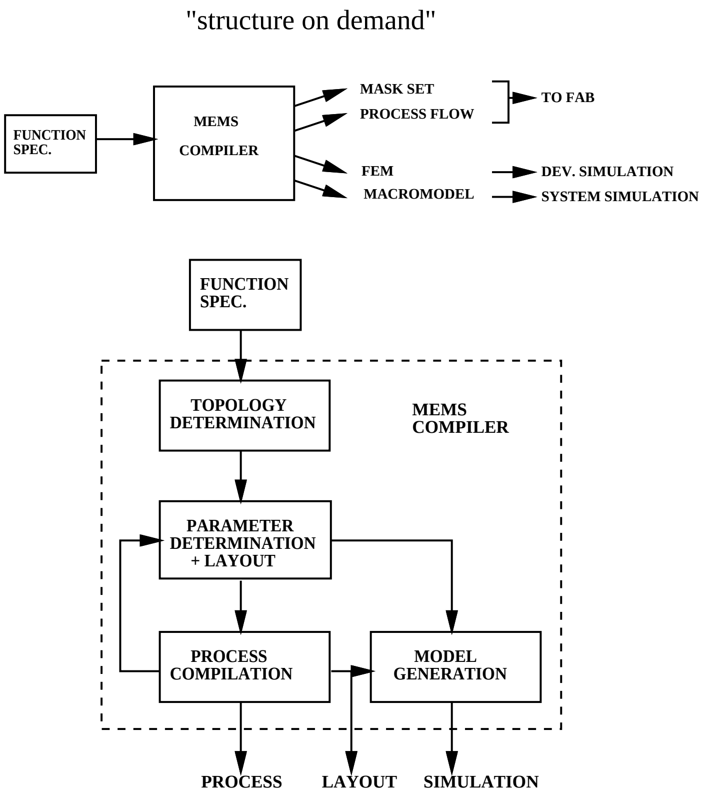


Figure 4.8.2: Top-down automatic design approach

References

[1] S. D. Senturia, “CAD for Microelectromechanical Systems”, invited talk at Transducer’s 95. June 25-29, 1995, Stockholm, Sweden. Vol 2, Paper No. 232-A7.

[2] R. A. Buser and S. B. Crary, “Integration of the Anisotropic Silicon Etching Program ASEP within the CAEMEMS CAD/CAE Framework”, in MEMS’92, 1992, pp. 133-138.

[3] H. U. Scharzenbach et. al., “A Microelectromechanical CAD Extension to SESES”, J. Micromechanics Microengineering, 3, 1993, pp. 118-122.

[4] M. Hasanuzzaman and C. H. Mastrangelo, “MISTIC 1.1: A Process Compiler for Micromachined Devices”, in Transducer’s 95, June 25-29, 1995, Stockholm, Vol. 1, Paper No. 38-A4.

[5] IntelliFab and IntelliCAD, IntelliSense Corporation.

[6] M. P. Bendsoe and N. Kikuchi, “Generating Optimal Topologies in Structural Design using a Homogenization Method”, Computer Methods in Applied Mechanics and Engineering, 71 (1988), pp. 197-224.

[7] D. C. Van Duyn, “Multi Signal-Domain Modelling of Solid-State Transducers: A Theoretical Framework Based on Irreversible Thermodynamics, Mixed Finite Elements and Multigrid”, Ph.D. Dissertation, Delft University. The Netherlands, 1993.

4.9 A Language for MEMS

Amar Mukherjee
Dept. of Computer Science
University of Central Florida
Orlando, FL 32816
`amar@cs.ucf.edu`

This Workshop is the third in the series of Workshops sponsored by NSF on design methodologies for VLSI-like technologies. First, I would like to present a brief summary of ideas on design hierarchy that were discussed at the prior Workshops. I will then outline a set of requirements for a possible layered language to describe MEMS structures.

Structured Design Methodology

Development of a structured design methodology played a key role in the VLSI (Very Large Scale Integration) revolution. In this methodology, the operation of the circuit is abstracted at different levels by formal systems, which allows separation between system design, component design and fabrication. The key element that contributed to this clean digital interface was the discovery of a set of scalable design rules (Mead and Conway,1980) allowing processing steps to be defined independent of object's geometry. For mechanical and electromechanical systems, in general, multilevel design hierarchy does not work very well because the underlying models involve energy transformation and physical parameters; the elemental components share function and behave differently in a system due to back loading (Voelcker, 1994 and Whitney, 1994). Even for VLSI, the traditional design hierarchy (system, function, logic, circuit, layout and fabrication) breaks down if we bring in performance considerations like power, timing, speed and chip area (Mukherjee, 1986). The design may need multiple iterations to take into account the side-effects of specially designed leaf cells which influence all the layers above. In fact, the exact behavior of the device can only be described by an analog model. For mechanical design, the performance considerations are an integral part of the design process. The shape and geometry of the part can be described by a geometry modeler but its functional behavior cannot be guaranteed unless the material satisfies certain physical properties and its geometrical dimensions satisfy certain size constraints depending on the application and performance requirements. The limits on geometric dimensions of the object and physical attributes of the material as they relate to correct function and performance of the object will be called design constraints. The design constraints are process independent and can be derived by experimental methods and mathematical modeling. This terminology is adopted in order to distinguish between factors affecting functionality and the manufacturability of the part. In the context of VLSI design, these two considerations can be merged into a set of conservative geometric design rules. An analogous set of design rules must also be satisfied by the mechanical fabrication process.

The design hierarchy consists of levels of abstractions, data exchange languages and digital interfaces as shown in Figure 4.9. The design subsystem performs the traditional design by deriving the shape and geometry that achieves the desired function and specification. The design must satisfy a set of design constraints with respect to a set of relevant mechanical and physical properties of the material. The analysis and simulation tools verify the correctness of the design. The design may go through multiple refinements before being delivered via the three-dimensional digital interface (Sproull,1994) to the physical design layer.

The design languages to be used for data exchange at the digital interfaces are areas of active research. A proposal for a new language, called SIF (Solid Interchange Format) has been advocated by Sequin at the last NSF Workshop (Sequin and McMains, 1995). Such a language should probably be based on a solid modeling system such as CSG (Constructive Solid Geometry) or BREP (Boundary Representation), possibly augmented by non-uniform rational B-spline (NURBS) surfaces and should form the basis of a standard 3-D data exchange language.

The physical design phase uses specific knowledge of the process and its design rules to specify a 2.5-D description of the part. Ideally, like in VLSI which satisfies layering paradigm with conservative design rules, this stage should be insensitive to the object's geometry. As is well known from studies of SFF processes, such a layering paradigm does not work in practice for objects with undercuts, objects with multigraded material or with material having anisotropic density. Furthermore, the smooth three dimensional surface features have to be compromised by linear approximation of zigzag surface features and additional constraints are put if the object needs support structures or overhangs.

The translation of the 3-D geometry to 2.5-D geometry has to be done by a 'slicer' that will produce the 2.5-D layers, given its description in SIF. A language called L-SIF (layered solid interchange format) has been suggested for this layered description but no specifics have been provided for any SFF process (Sequin and McMains, 1995 and Finger et al., 1995). This description will form a 2.5-D digital interface between the physical design and the process planning stage.

One key software is a design rules checker. The purpose of the design rules checker is to ensure the feasibility of the part fabrication. The tolerances of the fabricator must be reflected in the translator output of the slicer.

For VLSI design, the 3-D and the 2.5-D digital interfaces merged into one interface, the design rules allowed a clean separation between design and fabrication hiding the process planning steps in the set of masks required for a standard process. For MEMS, the processing steps depend on object's geometry.

The Language L-SIF (Layered Solid Interchange Format)

Surface micromachining uses a lyering paradigm that comes close to the VLSI layering and evolved with a strong motivation to use the VLSI fabrication facility in "as is" condition. In fact, it is simpler than VLSI since the process does not have to produce "active" components like transistors nor does it have to take into account the electrical parameters (the so-called "parasitics") that influence the connectivity considerations and the performance of the chip. Essentially there are three kinds of material that need to be layered to produce the desired

```
                            |
                     o-------------o
                     |   DESIGN    |
                     o-------------o     Functional
                            |            Design
                     o-------------o
                     |  FUNCTIONAL |
                     |  SIMULATION |
                     o-------------o
  3-D INTERFACE             |
=======================================================
  SIF (Solid Interchange    |
       Format)       o-------------o
                     |    SLICER   |
                     o-------------o     Physical
                            |            Design
                     o-------------o
                     | DESIGN RULES|
                     | VERIFICATION|
                     o-------------o
   2.5-D INTERFACE          |
 =======================================================
   L-SIF (Layered Solid     |
   Interchange Format o------------o
                      | PROCESS    |
                      | PLANNING   |
                      o------------o     Fabrication
                            |
                      o------------o
                      |FABRICATION |
                      o------------o
                            |
```

Figure 4.9: Design Hierarchy

parts: the structural layer, the sacrificial layer and the insulation layer. A different set of considerations such as deformation due to residual compressive stress, frictional properties and coupling phenomena such as stiction or bearing clearances come into the picture. If the domain of geometry is restricted to simple structures with a set of special standard constructs like bushing, bearing, cantilever, gear, rotor etc., a simple geometry language like CIF will serve the purpose. For bulk micromachining, several complicating factors arise which do not connect the layered description of the structures with the underlying process steps. The geometry for bulk micromachining seems to be the result of a sequence of anisotropic etching and etch stops that are custom made for the device. A geometric representation of a diaphragm, a hole, a cavity, a pin or a cantilever can still be defined with an implied semantics of how anisotropy is used to produce such a structure. Such special constructs have also been used for VLSI; ’butting’ or ’buried’ contacts or ’via’ or even the I/O pads are examples of such special constructs.

To incorporate arbitrary 3-D geometry, a more powerful layered language, L-SIF, will be necessary. Current implementation of surface micromachining such as MUMPS use CIF/GDS format or some variants or IGES format which can only handle polyhedral representations of solid. Special cell libraries of frequently used parts with non-polyhedral surfaces have also been provided in some instances. There is a need to develop a standard layered language, called L-SIF, for MEMS.

L-SIF will include the basic geometric constructs of CIF (Mead and Conway, 1980) which are:

Polygon with a path.
Box with length, width,center and direction.
Round flash with diameter and center.
Wire with width and path.

The layer attribute of CIF (which identifies polysilicon, metal, diffusion etc.) will be the same in L-SIF identifying the material of the layer in question. Since L-SIF will handle more complex geometry than CIF (which only handles Manhattan blocks and cylinders), L-SIF will have primitives that capture any slice of a quadric body such as sphere, ellipsoid, cone, paraboloid and hyperboloid with arbitrary orientation of the plane that cuts the slice. Extension of the language to cover bodies expressed by NURBS is also possible. The language will include constructs like:

[Layer name, quadric object, vertical resolution, orientation vector, slice number]

The object will be defined by its mathematical parameters. For example, for an ellipsoid we specify the coordinate of the center and the distances of its three radii. The orientation vector specifies the three components of a vector that defines the object’s rotation around the three major axes x,y and z. After placing the object on a horizontal plane with this orientation, imagine slices being cut in the vertical direction with the resolution factor provided and the slice number identifies which cut is the present cut being used.

The language should be able to describe layers with internal hollow regions. The "concentric" holes is defied as

[Concentric: label, scale factor]

where "label" inside the square bracket identifies a slice as before and scale factor defines a scaled down version of the object. The object consists of the labeled object from which the scaled down portion has been removed. The Boolean operations on layers like union, intersection or difference are defined as usual.

The L-SIF language will also have some of the general features that are allowable in CIF such as symbol definition, call, delete and user extension, comment etc. and the usual syntax of data types.

CIF is a standard data exchange language that can be mapped to different target technologies. Similarly, L-SIF should become a standard data exchange language for all SFF processes including MEMS which can be mapped to specific target technologies like surface, bulk or LIGA processes with its associated set of "libraries" of special constructs.

References

[1] Mead, C. and Conway, L., 1980, "Introduction to VLSI Systems", Addison-Wesley.

[2] Mukherjee, A., 1986, "Introduction to nMOS and CMOS VLSI Systems", Prentice Hall.

[3] Mukherjee, A. and Hilibrand, J., eds., 1994, "New Paradigms for Manufacturing", MIPS Division, CISE, National Science Foundation, Arlington, Virginia, May 2-4, 1994.

[4] Sequin, Carlo H. and McMains, Sara, "What Can We Learn from the VLSI CAD Revolution?", paper presented at the NSF Workshop on *Design Methodologies for Solid Freeform Fabrication*, Carnegie Mellon University, June 3-5, 1995.

[5] Sproull, R. F., "Digital Interfaces to Fabrication", *Proc. Workshop on New Paradigms for Manufacturing*, NSF, Arlington, VA, May 2-4, 1994.

[6] Voelcker, H., 1994, "New Paradigms for Mechanical Design & Manufacturing", *Proc. Workshop on New Paradigms for Manufacturing*, NSF, Arlington, VA, May 2-4, 1994.

[7] Whitney, D. E., 1994, "Some Differences Between VLSI Fabrication", *Proc. Workshop on New Paradigms for Manufacturing*, NSF, Arlington, VA, May 2-4, 1994.

4.10 A Case for Involving EDA Companies in MEMS CAD and Modeling Development Programs

Peter T. Parrish
Tanner Research, Inc.
180 North Vinedo Avenue
Pasadena, CA 91107
Peter.Parrish@tanner.com

The most important contribution that an EDA company can make in the area of MEMS CAD and Modeling is to productize the algorithm and technology development of universities in a form that gains industry acceptance for its commercial quality, completeness, and on-going support and development. In doing so, EDA companies seed the growth of an emerging technology in a number of areas that universities are not prepared to address. Tanner Research is uniquely positioned EDA supplier with a track record of successfully commercializing university research.

(1) Leverage Existing Software Tools A new technology is typically supported by a rough adaptation of pre-existing CAD tools. These adapted tools lack integration and features that ease transition to and adoption of a new technology. Developing these needed additional features in a widely-available, industry-tested tool, with a “bottom-up” analysis of the technology-specific issues and design flow is a much more cost-effective approach. By working with EDA companies, universities can devote their resources to algorithm development instead of duplication of existing software features. Work can be coordinated via EDA industry standard file formats and interface standards. Established EDA vendors can provide “hooks” into their programs that provide universities the opportunity create custom capabilities that build on the existing tool features. A good example of this type of hook is the User Programmable Interface (UPI), that Tanner Research provides with L-Edit^{TM}, for algorithmic creation of designs.

(2) Ease-of-Use and Technical Support Commercial tools consist of and imply much more than point tool features. Most tools have developed a graphical user interface, an established engineering design flow, and other added features that make it easy for users to enter, document and maintain designs; perform analysis; and visualize results. These tools also have extensive on-line help and documentation. Trained technical support staff are available by phone, fax and email. Support can also include application notes, tutorials, short courses, and other training materials.

(3) Maintenance and Platform Support Without clear technology-specific documentation, the barrier to the use of new technology is raised. As technology evolves, the process of updating and verifying the related software components, such as libraries and DRC/ERC rule sets, can become a major burden. EDA firms are prepared to track and respond to these changes. In order to reach the maximum market, EDA vendors support multiple platforms (e.g., PC-DOS, Windows, Macintosh and UNIX) and track platform and operating system changes. Multiple platform support is usually not a cost effective option for universities.

(4) Functionality and Quality In addition to “new features” there are “improved features”. Commercial tools are continually “fine-tuned” to increase solution speed, improve convergence, and decrease memory requirements. Commercial code is also subjected to a level of testing, QA and customer feedback that far exceeds what a university effort can justify, but which in the long term assures a high level of quality in the user’s hands.

4.11 Structured Design for MEMS

G. K. Ananthasuresh and Stephen D. Senturia
Microsystems Technology Laboratories
Massachusetts Institute of Technology
Cambridge, MA 02139
suresh@mtl.mit.edu
sds@mtl.mit.edu

Preamble

The following premises articulate our position on structured methods for design of MEMS.

1. Mechanical systems do not have the same modularity and topological simplicity as electronic circuits, and hence, pose a more difficult challenge to the development of systematic methods for design and manufacture. Therefore, caution should be exercised in comparing MEMS design with VLSI designs. Nevertheless, we see opportunities in MEMS for structured design, provided that the design domain is suitably restricted.

2. In contrast to macro systems, MEMS devices use fewer rigid link mechanisms and more intrinsically compliant components. Compliant mechanisms lend themselves to structured design, as it is possible to systematically generate these structures for specified controlled motion and force transmission [1, 2].

3. MEMS devices fabricated with planar lithographic processes share many features with VLSI devices. Hence, the mask design within a given process sequence may be a fruitful area for structured design. Similarly, it may be possible to design process sequences to achieve desired cross-sections [3].

4. Analysis is an important step in the design process. Unlike the integrated circuits field, which can draw on extensive sets of design rules and programs which automatically test for design-rule violations, the MEMS field lacks design verification tools at this time. One way to verify a design, prior to fabrication, is through numerical simulation. The strong coupling between different energy domains makes it extremely difficult, at present, to analyze a MEMS device using the existing simulation tools. Hence, there is a great need to develop computer aided analysis tools and system level simulators that are easily usable by the MEMS community[4, 5].

Discussion

Table 1 contrasts different aspects of VLSI systems, macro (traditional) mechanical systems, and MEMS.

Structured design methods exist for VLSI systems, because they operate in a single energy domain, involve a small set of primitives that possess a direct mapping between

VLSI Systems	*Macro-mechanical Systems*	*MEMS*
Single energy domain	Multiple coupled energy domains	Multiple coupled energy domains
Small set of primitives; decomposition is easy	Wide range of unstructured non-modular elements; hierarchical functional decomposition is not easy	The range of elements is not as broad as macro systems; scope for limited decomposition
Elements are clearly distinguishable functionally and topologically	Intrinsically shared topological boundaries. No direct mapping between function and form	Same as macro systems; topological segmentation is equally hard (e.g., a fluid volume bounded by moving parts)
Simple interconnection rules (KVL & KCL) among the decomposed elements	Interconnection rules are complex	Same as macro systems
Geometry of physical artifacts is not a big issue in design	Geometry of artifacts is intrinsically tied to the function they perform. Kinematics plays a big role	Same as macro systems, but kinematic issues are not as complex at present. Predominantly monolithic compliant structures
Manufactured with planar lithography	Wide range of manufacturing techniques including 3-D machining	Same as VLSI systems

Table 1. Comparison of VLSI, macro mechanical systems, and MEMS

function and topology, and have simple interconnection rules (KVL & KCL). In contrast, the macro mechanical field is so diverse in its designs and manufacturing techniques that it is not easily amenable to systematic design. It involves multiple energy domains and a wide range of unstructured primitives with intrinsically shared topological boundaries (e.g., a fluid volume bounded by moving parts). Furthermore, there is no clean correlation between the physical form of the artifacts and the function they perform. MEMS devices resemble macro systems in these aspects and retain some of the complexity and unstructured nature of macro mechanical systems. However, as can be seen in the last two items in the table, there are some features that make MEMS suitable for structured design. If limited to planar microfabrication processes and the types of MEMS devices available today (which require only a limited range of motion that is easily achieved with deformable structures), the geometry and kinematic issues are less complex than for macro mechanical systems. Another feature of MEMS that facilitates systematic design is the use of photo-lithography in microfabrication. The masks and the process sequence used in microfabrication provide a common interface between both the a description of the design and the recipe for manufacture. Therefore, there is an opportunity for the development of mask synthesis and/or process synthesis programs to assist structured design.

Systematic Design of Compliant Mechanisms A close look at the MEMS devices available today reveals that there is a paradigm shift from jointed rigid mechanisms to compliant mechanisms which are essentially deformable structures. It is possible to obtain, from functional specifications, conceptual designs for rigid and compliant structures systematically to support loads, and for controlled force and motion transmission, respectively [1, 2]. This has been done by suitably extending the already developed structural optimization techniques. Such methods require nominal input from the user in the form of functional specifications that include input-output forces and displacements, and have the potential to generate complete designs with enough details to proceed with mask generation automatically. Therefore, this is an area where structured design is possible.

CAD Tools for MEMS Analysis is an important component of design. Computer aided tools for analysis lead to systematic design methods. Due to the present lack of simulation tools, the only way MEMS designers can verify their designs is by building and testing, which is expensive and time consuming. For this reason, it is extremely important to be able to perform accurate simulations, accurate both in the geometric representation of the structure, and in the underlying constitutive properties and behavioral models for the MEMS device and the associated electronics. MEMS devices of today, most of which are transducers, operate in multiple energy domains, and there is a strong coupling among these domains (e.g., coupled electro-mechanics, fluid-structure interaction). This requires specialized analysis methods for accurate 3-D simulation [5]. Furthermore, the backgrounds of MEMS designers are very diverse because of the multi-disciplinary nature of the field. Consequently, it is important to develop the analysis tools in such a way that most MEMS designers can use them comfortably. For instance, if electrical engineers need to do finite element analysis of the mechanical structure or if mechanical engineers need to simulate an electronic circuit, there should be suitable interfacing tools.

System Level Modeling A MEMS system typically contains a variety of devices including electronics. A common representation that encompasses multiple energy domains is useful in modeling the whole system. The bond-graph notation, which is based on energy transport (or power flow) may prove to be useful in representing the entire system at the highest level. Ultimately, one seeks the dynamical behavior of the entire system. But most transducers are nonlinear, involve at least two energy domains, and operate in the large signal regime. Direct numerical simulation of the dynamics of the fully meshed distributed model of the system is computationally difficult and is very expensive. Therefore, it is necessary to reduce the number of degrees of freedom from the hundreds or thousands of degrees of freedom of the meshed 3-D model to as few as possible. Such reduced order models can then be used for system level dynamic simulation [4]. These ’macromodels’ should be developed in such a way that they agree with both 3-D numerical simulation and experimental results in describing the macro behavior of the system. Macromodels can also be used to represent the behavior of a subsystem within one energy domain, or the interaction from other domains. Hence, an important goal in MEMS design is to develop means to automatically generate macromodels and insert such models into a system- level dynamic simulator.

Conclusion

A cautious approach should be taken in trying to establish structured representation and design rules for micromechanical systems, as mechanical systems are known to be not amenable to either systematic design or clean separation of design and fabrication. In spite of difficulties, there are some avenues in the MEMS area for structured design such as mask and/or process synthesis for a selected class of geometries, synthesis of compliant structures, and computer aided tools for quick and accurate analysis of different classes of MEMS devices.

References

[1] Bendsoe, M. P., and Kikuchi, N., “Generating Optimal Topologies in Structural Design Using a Homogenization Method”, Computer Methods in Applied Mechanics and Engineering, 71 (1988), pp. 197-224.

[2] Ananthasuresh, G. K.; Kota, S.; and Gianchandani, Y., “A Methodical Approach to the Synthesis of Compliant Micromechanisms”, Technical Digest, 1994 Solid State Sensor and Actuator Workshop held at Hilton Head Island, SC, June 13-16, pp. 181-185.

[3] Hasanuzzaman, H. and Mastrangelo, C. H., “MISTIC 1.1: Process Compiler for Micromachined Devices”, Transducers ’95, June 25-29, Stockholm, Sweden, Vol. 1, Paper No. 38-A4.

[4] Senturia, S. D., “CAD for Microelectromechanical Systems”, Invited Talk at Transducers ’95, June 25-29, Stockholm, Sweden; also appears in the proceedings, Vol. 2, Paper No. 232-A7.

[5] Gilbert, J.; Legtenberg, R.; and Senturia, S. D., “3D Coupled Electro-mechanics for MEMS: Applications of CoSolve-EM”, Proc. IEEE Workshop on Microelectromechanical Systems, MEMS ’95, Jan. 27-Feb. 2, Amsterdam, Netherlands, pp. 122-127.

4.12 Thoughts for the MEMS Community

Carlo H. Sequin
CS Division
U.C. Berkeley
sequin@eecs.berkeley.edu

Here are some additional thoughts to accompany the paper: “VLSI Design and Fabrication” by Carlo H. Sequin and Sara McMains. These thoughts are aimed particularly at the community of MEMS designers and fabricators who will come to the Caltech workshop.

The Problem Domain

MEMS occupy an interesting niche between VLSI chips and miniature mechanical systems. Their structures can be far more complex in the vertical dimension than the VLSI circuitry used in computer or memory chips. No good widely accepted language currently exists to describe the 3D geometry of the finished structures. The complete set of all mask geometries together with the travel sheet that describes the process sequence, at best give an implicit description of the resulting product. From this implicit description, it is in principle possible to obtain an explicit description of the geometry of the finished chip if suitable modeling programs are available for all processes and a suitable geometric modeling language is used.

A traditional way to describe mask geometry in the field of VLSI is through the use of some mask geometry language such as CIF. Some mask description formats have been provided with extensions that allow to describe in more details the thicknesses and properties of the various layers in different regions. Most of these extended formats are still 2.5D description formats, rather than real 3D geometry formats. They are thus inappropriate to describe complicated overhangs or features that occur on vertical faces.

One could use some established solid modeling language (e.g., ACIS) to describe the geometry of the chip at various states during the sequence of process steps needed for manufacturing. However, commercial solid modeling languages have the wrong strengths. They provide compact high-level ways to define a desired solid shape, often focusing on constructive solids geometry (CSG) operations, but they have virtually no support for tracking a complicated manifold such as the exposed surface of a silicon dioxide layer as a function of time during the simulation of an etching process. What we need for the use with advanced IC chips and with MEMS is a descriptive geometry language that captures the state of a complicated physical structure consisting of several different regions with different materials properties, which can change as a function of time. The immediate needs are for primarily a descriptive language, rather than for a design language, since we are still far from the point where it will be possible to define an ideal shape or a complete mechanism, and then have it manufactured in a top-down manner by defining automatically the needed mask geometries and fabrication steps. For the foreseeable future, MEMS will have to be designed from the mask-set point of view based on a detailed manufacturing process plan.

There will not be a single generic MEMS process! Every new application, and every new conceptual structure may need a new process sequence for its implementation.

Relations between CIF, SIF, L-SIF, and UniGrafix

Berkeley UniGrafix, is a simple geometric description language that we have been using in computer graphics research and instruction since the early 1980's. It was created as a 3D extension of CIF, the Caltech Interchange Format, for situations where a 2.5D layered description is no longer appropriate. It is almost a strict superset of the 2D CIF language, but some of the primitives specifically aimed at the needs of IC layout, such as "path", have been left out of the 3D language and have instead been relegated to the domain of a separate set of generator tools; e.g., the "UGworm"-tool will generate a properly mitered prismatic tube along a piece-wise linear path through 3D-space. Berkeley UniGrafix would be quite appropriate for the description of even the most complicated, static 3D MEMS geometries. It does not have adequate facilities, however, for describing shapes that change as a function of time.

We are now in the process of defining SIF, an interchange format for Solid Free-Form shape manufacturing (see article referenced above). Thus the question naturally arises: Could this also be the language of choice for the MEMS community ? The answer is: Probably not ! The needs of the two communities are too different. The SFF world needs a language to capture designs of 3D shapes. The MEMS world needs to capture descriptions of complex structures, which, however, may be time-varying. The L-CIF language is closer to the descriptive end of the spectrum, but it is again a layered language and thus does not overcome the key shortcomings of the extended 2.5D mask description formats that are used today.

In my opinion, the ideal MEMS description language would follow a strict boundary representation paradigm and be efficient to represent many extended complicated surfaces with many detailed features, probably in the form of a shared vertex array and a triangular or quadrilateral mesh referencing these vertices. It must have the provisions to attach reasonably complex volume specifications to any of the regions bounded by a combination of parts of these surfaces, e.g., a gradated doping in the z-direction. As in all interchange formats, it is absolutely vital to have a cleanly defined semantic meaning for all expressible constructs; syntactic choices are more a matter of taste.

Other CAD Tools

MEMS may have moving parts, e.g., electro-motor rotors that turn or levers that flip up after processing is completed. Representation of such operations will require the power to formulate 3D rigid body transforms. To be further able to do at least a kinematic simulation of such MEMS operations may require even more sophistication; since these motions are typically constrained in some way, a complete constraint solver system may be required.

An important CAD operation that is performed on a description of the expected geometry of a processed device is "extraction". This is the process of finding a higher-level description from a large – possibly unstructured – mass of lower-level primitives. In the IC domain, extraction is often used as a verification step after a lengthy multi-stage layout compilation to verify that the sum total of all layout rectangles indeed constitutes a circuit with the desired properties.

For a MEMS with an interesting 3D geometry such extraction is much harder than for the typical digital circuit chip. There may not be a rich enough set of predefined higher-level primitives in which the function of the MEMS structure can be expressed succinctly. The richness of the possible geometrical structures that can be formed is far too rich.

The extraction, simulation, and analysis of MEMS function are a challenging research domain that will not likely yield to some standardization in the near future. Efforts should thus best be spent towards achieving a lower goal, i.e., that of creating a simple geometry language extension that allows to adequately describe resulting or desired MEMS geometries. An extension of 2D CIF to a true 3D language in the spirit of Berkeley UniGrafix may be all that is required.

4.13 What Can We Learn from the VLSI Revolution

Carlo H. Sequin and Sara McMains
CS Division
U.C. Berkeley
`sequin@eecs.berkeley.edu`

1. VLSI Design and Fabrication

Before forging ahead with new standards for SFF, it seems worthwhile to look at design and fabrication in the domain of integrated circuits and to review the VLSI CAD revolution of the 1980s. There may well be some valuable lessons that carry forward to the domain of SFF.

IC fabrication is a 2.5-dimensional technology. There are many different classes of processes: NMOS, CMOS, Bipolar, BiCMOS, GaAs .... In each class there are many variations among different fabrication lines. But in all cases the principle is the same: one starts with a flat wafer, usually a silicon crystal lattice with a single orientation, and then produces a series of changes in areas that are defined by the geometrical patterns on a sequence of fabrication masks. In order to produce a "Si-gate NMOS" circuit (a vertical layering of Metal and Oxide, on top of a Semiconductor which has regions of Negatively charged carriers), one might use the following key conceptual steps. One mask pattern (often symbolically shown in green color) and the associated processing steps may cut open a pattern of trenches into the oxidized silicon crystal surface, into which small amounts of "impurities" are diffused in order to produce regions that are more conductive than the bulk of the silicon substrate. A second (red) pattern may define connection strips in a deposited polycrystalline silicon (Poly-Si) layer on top of the previous structure. Where these strips cross the "green" trenches, transistors are formed: the voltage on the red strips – or "Si-gates" – controls the amount of conductivity in the green trenches underneath. A fourth (blue) pattern defines another layer of interconnections, in a metal layer on top of the whole processed wafer. A third (black) pattern specifies where connection holes should be cut through the insulating layers placed between the "red" and "blue" interconnection levels.

For a given class of processes, the basic functionality of the various layers is always the same, even though the thicknesses of the many layers and the exact cross-sectional shapes of the trenches and conductor strips may vary. In a different process class, the levels may perform different functions; for instance, in an older "metal-gate NMOS" process, the amount of conductivity in the semiconductor bulk layer is controlled by gates formed in the metal layer. Whoever designs the masking patterns for a particular circuit has to understand what the basic functions are of the various patterned layers. In the early 1970s the design of a circuit layout was an "art" where each designer had to work closely with the people operating the fabrication line in order to define appropriate patterns that together with the envisioned processing sequence would then produce the desired circuit behavior.

One of the contributions of Mead and Conway ["Introduction to VLSI Systems," Addison Wesley, 1980] was to push for a simplification and standardization of this interface between designers and fabricators. For many applications, the performance of the digital logic circuits is not very critical compared to the capabilities of the technology and will be good enough even when the potential of the chosen fabrication process is not used to its limits. It was thus possible to define a simple set of geometrical layout rules that would assure that functional transistors and circuits of acceptable performance would be produced for fabrication lines whose processing parameters were kept within reasonable parametric ranges.

This was done first for the Si-gate NMOS process and later for “CMOS” technology (Complimentary MOS, with transistors of both polarities). This simplification allowed an abstraction of the design process, where geometrical “recipes” would produce transistors of desired performance (speed, driving power) or circuits of desired functionality (AND-gates, NOR-gates, multiplexors . . .).

To make this abstraction work, it is crucial that the semantics of the various patterns is well defined. After some initial controversy, the geometrical shapes were defined to represent “what you can see when you look down on the final IC chip”. Since the actual masks used in fabrication may actually have different dimensions, owing to expansion or shrinking of the various features during the fabrication process, or owing to size changes when masks are copied, it became the responsibility of the fabrication houses to compensate for all these cumulative effects and to produce the masks for their own use that would deliver the specified geometrical dimensions on the final chip. Thus the fab-line dependent variations and idiosyncrasies became invisible to the circuit designer, and the same set of geometrical pattern specifications became portable from one fab-line to another. As technology improved and it became feasible to make ever smaller features and thus faster and more compact circuits, it was even possible to take the old patterns and to shrink them automatically to make them usable again for these advanced processes. At this point one has reached a true abstraction of the layout design process, where the designers may not even know what the dimensions are of the rectangles that they are drawing.

2. Higher Levels of IC Abstractions

After the lowest level of abstraction of implementing pieces of integrated circuits (by specifying the raw geometry of all the features on the chip) became routine, and ever larger collections of useful and reliable circuit elements had been compiled, it became practical to shift the level at which designers would think about their circuits to a higher level. The logic level had already become well established with the library of bipolar circuits described in Texas Instruments’ famous TTL Data Book. The semantics or the functions of the various gates was quite clear, and library cells that implemented these various useful logic circuits could readily be designed and optimized for different fabrication processes. The designers now simply had to plug together these “logic gates” into more complicated digital subsystems.

Very soon, frequently used higher-level circuit components emerged: registers, datapaths, memory blocks, arithmetic-logic-units (ALUs); these became the natural – and often parameterized – building blocks for larger systems such as processor architectures. This "register-transfer-level" of abstraction is most conveniently used when a designer tries to trade off system speed against implementation costs and tries to find the most cost-effective degree of parallelism in the architecture.

Between the layout level and the logic level, there are a couple of other abstraction levels that were often used by many designers. First there is the "sticks" level in which the connections in the various layers of the integrated circuit are not represented by their exact geometry, but only by their topological ordering and arrangement on the chip surface. This level can be drawn more quickly and frees the designer from worrying about exact dimensions and about layout-rule correctness. A computer program, called a "compactor," then converts a sticks diagram into a densely packed, layout-rule correct geometric layout. However, this level never became useful for exchanging designs between designers in different groups, since the semantics of this description was never clearly defined and standardized.

Similarly, yet another abstraction level, the "switch" level, never played a major role as an interchange standard, even though this level is very important for circuit simulations and for optimizing the performance of a generic circuit. Too many parameters would have to be specified with each component at the switch level, and the actual performance would then still be dominated by many parasitic circuit elements, such as stray capacitances that are strongly affected by the exact layout and by the relative arrangements of the circuit components on the chip.

Above the register transfer level, there are also further levels of abstraction. But as one climbs higher up in this abstraction hierarchy, the concerns of the designers become richer and harder to define. At the "functional" or at the "behavioral" levels, one tries to formally capture what the system overall is supposed to do – but this is rather difficult. For a long time, certain semantic issues had been left unanswered: for instance: "What is the meaning of time in this domain? Is it continuous or discrete? Is it fully synchronous to some clock? And if so, how are external asynchronous events handled?" The lack of semantic clarity and completeness made it difficult to exchange systems specifications at this level. In the last few years, however, a standard language (VHDL) with reasonably well defined semantics has emerged.

3. Conversion and Checking Tools

Defining these languages and interchange formats with proper semantics is just the beginning; they are not much use unless we also have convenient tools to create descriptions at these various levels and to convert these descriptions from one level to another.

Programs that convert descriptions from higher levels of abstraction to lower, more detailed description levels are known as "generators" or "compilers". For example, we might have a parameterized "N-bit full adder" generator, which – upon specifying a value for N – will generate an adder circuit with N bits in a particular technology. This relieves the individual designer of the tedium of explicitly specifying all the geometric details of the layout of such a circuit.

The inverse process, finding a higher-level description from a large – possibly unstructured – mass of lower-level primitives is known as "extraction". This is often used as a verification step after a lengthy, multi-stage compilation process, e.g., to verify that a large collection of layout rectangles indeed constitutes a circuit with the desired properties. Such a circuit-extraction process may also find spurious circuit elements that never show up explicitly in the generation process. For instance, long interconnection lines may have associated with them a significant amount of capacitance to the substrate. To adequately model the performance of such a "transmission line," this capacitance has to be calculated and suitably incorporated into the model of the circuit.

To close the verification process, the extracted circuit would have to be compared to the original, intended circuit. This can basically be done with some graph-matching algorithm; however, some margin of error has to be tolerated in the parameter values for each individual circuit component because of the parasitic geometries mentioned above. There is another difficulty: the same flat circuit or logic description can often be hierarchically grouped in different ways that are equally valid or "natural". The extracted hierarchy may then be quite different from the one originally conceived by the designer, and thus hard to compare. It might thus be worthwhile to leave "hints" or other informal information in lower level descriptions that convey the original design intent.

There are also internal consistency checks that can be made on individual descriptions alone. For instance, in a circuit diagram, one can check that differently labeled signal or power lines are not short-circuited to one another, or that all terminals with the same labels are indeed internally connected. At the level of the layout geometry, one would typically apply some geometric checks to see that the chip can be fabricated with reasonable tolerances and will still yield – with high probability – the geometric patterns that the designer had in mind. Primarily, such layout checks would test for certain minimal separations between elements that should not fuse together, and for certain minimal internal dimensions for paths that should not be broken. Comparing the geometry on two different levels, one would check for sufficient tolerances in the mutual overlap of features, so that one retains coverage even in the presence of small mask misalignments. All these checks are tightly coupled to the semantics of the features represented and to an understanding of the capabilities of the processes that are controlled by these descriptions.

4. What Can We Learn from This?

Many similarities become apparent when looking at some of the SFF processes in the context of the above experiences with VLSI. Here we present our initial reactions to the questions raised in the call for participation for this workshop. In many instances where the situation in SFF is sufficiently different from that in the domain of VLSI, we are not ready to take a definite stand; we then simply present the pros and cons of an issue to help focus the discussion at the workshop.

4.1 Modeling and Design Exchange

> *What level of abstraction should be used for describing the physical design? Should it be two-dimensional layers corresponding to the layers which are built*

> *during the fabrication process? Or should it be a form of a three-dimensional description with features identified?*

A single level of abstraction will definitely not be sufficient. There is a need to describe mechanical systems and their physical parts at one or more high-level abstractions that capture various aspects of their functionality. These higher levels of abstraction are outside the concern of this workshop.

However, at some point, the shape of an individual part has to be defined, and a clean abstraction level is needed to describe the geometry of this part in a completely fabrication-process-independent way. This geometric shape specification should clearly be a 3D description. Many existing solid modeling languages would be quite adequate for this purpose. They should have the power and expressibility to describe smooth surfaces in a resolution-independent, non-tessellated, compact form, and should be able to represent cylinders and spheres perfectly. NURBS seem to be one of the obvious choices for that purpose.

However, the variety of existing solid modeling systems and the possible complexity of the constructs they offer may be bewildering and scary to the manufacturing people who should be able to accept and process models in this form. It might thus be worthwhile to define a lean and clean subset language that contains only the really essential elements and has unambiguous semantics. For discussion purposes we will call such a language “SIF,” or “Solid Interchange Format”. This should be a compact, human-readable ASCII language in the spirit of CIF. The format should be hierarchical to avoid the waste of transmitting repetitive data.

“Features” (e.g., a “hinge,” or a “parameterized dove-tail slot”) – like functionality – do not belong at the SIF level, but at a higher symbolic level. While including such features may make the job of the process planner and the rule checker easier (see Weiss and Prinz draft position paper), it may be too daunting to define a rich enough catalog of features for all current and future application domains. (Possibly, if a high-level description exists that includes features, this information might be passed down to lower level descriptions to make the extraction of fabrication features easier. But we would then need to discuss to what degree such “hints” are an informal local help and to what degree they should be officially supported in SIF.)

In general, any such language should be kept as simple as possible, but general enough to do all the jobs envisioned, and extensible in case new needs emerge that cannot be handled by the original format.

There is also a need for lower level part descriptions that are much closer to actual fabrication plans. For SFF, a natural candidate would be some 2.5D layered description that is comparable to the CIF description of mask levels for a particular fabrication process. For the sake of discussion, we call this format “L-SIF,” for layered-SIF. The L-SIF descriptions for one and the same part may be quite different for different SFF fabrication processes.

Ideally, L-SIF would have a very similar style and semantics as the SIF language and should in principle be a derivative thereof. On the one hand, it might contain fewer geometrical constructs than SIF, since it might only have to express the 2D shapes of subsequent layers; thus, rather than having to describe NURBS surfaces, it would only have to express NURBS contours. On the other hand, the L-SIF languages for different SFF fabrication classes may need to convey some extra information such as the direction of a material-depositing nozzle movement. For some processes, such as Shape Deposition Manufacturing, the L-SIF process plan description might require almost the same richness of geometrical information as the idealized SIF part specification, since the individual layers are so thick and individually machined around their perimeters that they must be considered 3D solids in their own right.

Finally, below the level of L-SIF, there will be a plethora of machine-dependent control languages over which our task force will have little influence.

> *What type of model should be used to represent designs? What should be the role of traditional solid modeling? What attributes should the model provide in describing the design other than geometry? Candidates include strength, material, tolerance, etc.*

Traditional solid modeling systems have mostly relied on polyhedral or on smooth "boundary representations" (B-REP) or on "constructive solid geometry" (CSG) assemblies of a few clean primitives (cubes, cylinders, spheres, . . .), or of a combination thereof. While other models exist for describing geometry that are mathematically purer, it is not clear that they are preferable to describe shapes at the SIF or at the L-SIF level or that they would be accepted by a generation of engineers that grew up with today's CAD and modeling tools.

For cleanliness and compactness of the description it may well be desirable to allow SIF descriptions to be a combination of B-REP and CSG: i.e., regularized boolean set operations trees whose leaves are B-REPs of 2-manifolds, half spaces, and partially bounded objects. This would allow, for instance, a compact, resolution-independent specification of a shape defined by one or more general NURBS with some cylindrical holes in it.

At the SIF level, the different geometrical regions defined would have to carry information on the "desired" volume or surface properties, which the manufacturing processes would then have to approximate as closely as possible. Material selection, density, color, tolerances, and surface roughness may be such properties that need to be specified. This could be accomplished by using a volume statement that takes a solid region and specifies the "content" or materials properties of that domain. Here is an example that demonstrates the use of CSG and of several geometric regions occupying the same space: graded coloring could be specified in a large bounding box that contains the whole part and which is formally intersected with it.

At the L-SIF levels, the volume or boundary information may then be much more dependent on the particular SFF process envisioned and specifically refer to some of the available materials that could be dispensed in each layer, or to the particular local treatment that these layers may experience at certain locations.

Specifying the material at the SIF level raises some interesting issues. Since different processes have different material capabilities, a design that includes a material specification is no longer process independent. For a manufacturing process where SFF will only be the first stage of the tooling, such as making wax positives for investment casting, the material the SFF uses won't be the material of the final part. But for a different SFF process that can make parts directly out of metal, the SFF will use metal instead of wax to produce what will ultimately be the same part. (We should probably limit ourselves to specifying the properties and materials of the primary part emerging directly from the SFF process.

A key issue is that of auxiliary support structures that are not part of the final geometry but are required for a specific manufacturing process. For example, wax patterns for investment casting need to be attached to the sprue, gates, and runners in order to distribute the molten metal, while wax patterns that are to serve as conceptual models would not need this additional geometry defined. Similarly, a part produced by stereolithography (SLA) will need support structures for cantilevered portions of the model, whereas these would be unnecessary if the part were produced by selective laser sintering (SLS). Generating such "supporting" geometrical features is not yet fully automated. While the designer may have to think about the role that these supports play in her part design and how the supports will get removed and how the break-off ridges may get smoothed, the actual placement of the support geometry is normally done by the experts at the fabrication service, since they are much more knowledgeable about the needs for supports in their particular process than the designer.

Another issue that may get addressed quite differently at the SIF level and in the L-SIF descriptions concerns the way one may specify embedded components. At the SIF level, the designer may simply specify the location of the component and it could then be inferred that the embedding material has a corresponding cavity at that location; alternatively, a suitable cavity could be specified explicitly with a boolean difference operation. At the L-SIF level, the cross section of such a cavity may appear explicitly on each layer; alternatively, an explicit pause statement at the layer touching the minimum-location of the part to be embedded could be generated by the slicer program that creates the L-SIF description from the higher level SIF part description. The STL format currently has no explicit way to handle such embeddings, making it necessary for the designer to convey her intent to the fabrication house separately. The SLA machine operator then adds a "pause" statement to the control code at the layer where the embedded component will be placed, and then manually inserts the desired component.

> *What should be the form of a design exchange format? Can the format support alternative SFF processes? Is there a common set of information required by all SFF processes?*

Ideally, the exchange format should be at as high a level as possible and should be process and machine independent; this is the level of SIF which specifies "what" shape is desired, but ignores "how" this is accomplished. However, this approach is appropriate only if the worst case process inaccuracies are smaller than the finest feature that the designer is concerned about.

As the design specifications get closer to the limitations of the technology, the idiosyncrasies of the fabrication process will have to be taken into account. Today, most SFF processes have quite different capabilities, and most are not even isotropic, so that the part orientation during the fabrication process becomes a crucial issue. A particular SFF process will thus have to be targeted and parameters such as part orientation, layer thickness, and auxiliary process steps such as curing or shot peening may have to be specified explicitly. Under those circumstances, it may be more appropriate to send a part specification to the fab-house in the form of a process plan at the L-SIF level.

Suitable CAD tools must be developed to assist in this conversion process. The key is that each process and the corresponding conversion must try to approximate the desired geometry as closely as possible within the technological limitations.

Internally to the fab house, this L-SIF description may then be converted further into a sequence of instructions to drive a particular machine – again trying to realize the specified 2.5D geometry as closely as possible. This latter step should definitely be hidden from the client.

4.2 Design Rules and Tools

> *What type of design rules can be defined that when applied to the representation will guarantee successful fabrication in a series of SFF fabrication processes?*

At the SIF level, we may employ generic tests to check that the representation describes closed solid objects (as already exist for `.STL` files. An other test would be looking for intersections of solids having incompatible volume property specifications, e.g., different materials.

At the L-SIF level, at the latest, different design rules must be applied for each different class of SFF process – otherwise fabrication will be limited to the least common denominator of process capabilities. Rules could include the minimum separation of surfaces that will be guaranteed not to fuse (e.g., to make a tight bearing), and the minimum wall thickness for a given geometry that will not punch through or collapse (e.g., to make a wine glass); this minimum thickness may be different for different orientations.

For SLA in particular, design rule checkers should test whether all parts of an object are anchored to the build platform during all stages of the build process, and whether features have adequate supports. Any process subject to curl could have rule checkers to test whether the curl distortion of the part is predicted to lie within acceptable limits.

> *What should be the nature of design tools to support SFF? VLSI research produced design capture, design rule checking, mapping of logical constructs into physical transistors, physical placement of transistors, and wire routing. Are there analogous tools for SFF including design capture, design critics to identify non-manufacturable features, and fabrication process planning?*

Clearly we want as much automated generation and compilation as possible to speed up the overall prototyping process and enhance its reliability. We also need extraction tools and certain verification tools at all levels. Like the design rules themselves, some design tools will be process specific. Process planners in particular have very different requirements for the different SFF technologies, and very different process planners are shipped with the various commercial SFF systems.

Another question that needs to be addressed is at what point in the process should the orientation for the final build be chosen? Different SFF processes may have different properties along different axes of manufacturing. Definitely the axis perpendicular to any layering is special, but even the remaining two axes may have different properties in some manufacturing processes. Designers may or may not know about these differences. If they do not know, they will have to design for a worst case orientation making the weakest assumptions for all directions. If they are very knowledgeable about the process to be used and its performance characteristics in different directions, then they will want to specify the exact orientation in which the designed geometry should be built in the process framework. A useful area of research might be the development of process specific tools that attempt to find the "best" orientation for a part, if the designer doesn't know. For processes that can build multiple parts in a single build cycle, the "best" orientation may also depend on the spaces left by the other parts being built in the same build cycle. Whether specified by the designer or derived automatically, design rules, simulation, and process planning will all need to know the orientation.

5. Conclusions

In the domain of VLSI CAD tools, the vision has always been "top-down" with a focus on the "holy grail" of a fully automatic "silicon compiler". However, the suite of tools that has actually made a difference and that has revolutionized the world has been built "bottom-up," with a first focus on simple plotting, circuit extraction, and layout-rule checking tools.

This was followed by synthesis tools that could automatically generate some lower description from the next higher one, and several of which could eventually be cascaded in a sequence of compilation steps.

The final round then was to include optimization in this compilation step; it was no longer good enough to just produce an acceptable solution – one wanted to obtain the "best" possible solution at the lowest possible "price".

We are convinced that this same evolution paradigm will also apply in the domain of SFF fabrication.

4.14 Position Paper

John Tanner
Tanner Research, Inc.
180 North Vinedo Avenue
Pasadena, CA 91107
`John.Tanner@tanner.com`

Tanner Research aims to deliver tools and libraries to the electronics designer's desktop (network)—tools that are easy to use, and are available on the most popular platforms at affordable prices. Our goal is to provide MEMS designers with a powerful, highly integrated, state-of-the-art design tool suite by leveraging off of our existing VLSI CAD tools.

Our roots are in the Mead-Conway methodology for the design of VLSI. We have developed a suite of commercial IC design tools that are currently used by a variety of MEMS designers. Our IC and Multi-Chip Module (MCM) tools of interest to MEMS designers include:

- Hierarchical all-angle layer-based polygon editor
- Design rule checker (current version orthogonal—soon 45s)
- Device and circuit extractor (current version orthogonal—soon 45s)
- Interconnect only extractor (soon)
- Layout vs. Schematic circuit comparison
- Technology files and libraries for MOSIS/NIST processes
- Cross-section viewer
- Spice circuit simulator
- Schematic editor
- 3D finite-element based thermal analysis tool
- 2D boundary-element EM analysis tool

Under government funding, we are nearing completion of the following enhancements to our tools to better support MEMS:

- Arc and torus drawing primitives for the layout editor
- All-angle rotation of instances in the layout editor
- Variable etch angles within the cross-section viewer
- Technology files and libraries for the MUMPS process

- 3D boundary element EM analysis

Under government funding, we are beginning an investigation of the following features for MEMS:

- Use of interpreted C language for user-coded MEMS layout synthesis
- Use of interpreted C language for user-coded Spice electro-mechanical models
- Simplified automatic conversion of 2-D layout to 3-D representation
- 3D rendering
- Incorporating mechanical models into schematics and circuit simulation

In addition, we believe the following tools are needed:

- All-angle design rule checking and extraction
- Linked finite-element simulation of electric, mechanical, magnetic, thermal, and, for some applications, fluid properties
- High level model development (analytical, circuit level)
- Automatic model parameter extraction
- Advanced 2D to 3D conversion for MEMS-specific process steps
- Mechanical simulation with collision detection for hinged structures
- Design manager for organizing multi-disciplinary, multi-level design and model representations
- Lots of domain-specific model libraries from experts

We see similarities and differences between MEMS and VLSI designs. They both benefit from full hierarchical layout design and a mixture of hand-crafted cells and algorithmic code-based synthesis. MEMS needs a richer set of layout capabilities including arcs as drawing primitives, all-angle rotations, and all-angle DRC and extraction.

As others have noted, MEMS allows the design of a much richer set of devices than in the IC world where there are just a few kinds of transistors. Thus the tools for the extraction of netlists from geometry, tools that for VLSI can discover devices as well as interconnect, will be limited for MEMS designs to discovering the interconnect between predefined hierarchical cells. However, as MEMS designers move from a test/prototype environment to a production environment, tools such as our block extractor become important for back end validation.

As with VLSI, simulations of MEMS devices of significant size cannot be done at the finest detail within the limits of computing memory and simulation time available on the desktop (network). Simplifying abstractions must be made to reduce the simulation load while still maintaining the behavior of interest. In VLSI, levels of abstraction range from finite- element (TCAD) through circuit-level, gate-level, and behavioral (HDLs). For MEMS, levels of abstraction range from finite-element through circuit level (with mechanical behavior) to behavioral level.

In both disciplines, the simplifications come from abstracting away some significant level of detail. The designer is best at determining the essential desired behavior and thus higher levels of abstraction are almost always defined by the domain expert. Computationally intensive tools could be used to perform analysis and simulations, under the experimental control of the modeler, to verify that a more abstract model simulation matches close enough to the more detailed one.

We allow researchers and designers to extend our tools by writing their own C-language routines that link into our tools or are interpreted by our tools. This mechanism allows researchers to leverage our efforts to provide robust multi-platform base functionality while adding their own innovations with minimal overhead. This method of collaboration only works if we provide the right set of extension hooks into our tools.

Perhaps it is our VLSI bias, but we foresee a large and very interesting set of MEMS chips that will contain a small number of mechanical elements integrated closely with electronics. In this scenario, it is critical that the new MEMS aspects of design be incorporated into the existing, well-proven IC design tools and procedures. This view drives our development to add new MEMS capability to our integrated tool set.

At the upcoming NSF workshop, we hope to gauge the importance of the list of tools that we have identified above that are lacking, learn about other tool needs, and identify the areas that our tools need hooks for user extensions.

4.15 Coupled-Domain and Mixed Regime Numerical Techniques for Microelectromechanical Simulations

J. K. White and N. R. Aluru
Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology
Cambridge, MA 02139
white@mtl.mit.edu
aluru@rle-vlsi.mit.edu

Designers of VLSI integrated circuits use hierarchical or mixed level simulation that allow them to focus on the details of one section of a design, while still efficiently simulating the entire circuit. Such tools allow a relatively free mixture of behavioral, register-transfer level, logical, and circuit-level descriptions of a given design. For designers of MEMS, the need for the equivalent of mixed-level simulation is even more pressing than it was for VLSI designers, because simulating even a single device's performance usually requires a multi-level approach. For example, while using the MIT MEMCAD system to analyze the Analog Devices comb-drive based accelerometer, we used an *ad-hoc* approach to mixed-level simulation. The electrostatic force-displacement relation for each of the comb fingers was macro-modeled by table, and then this macro-model was combined with a 3-D mechanical model of the polysilicon proof-mass plug spring system.

The development of mixed-level MEMS simulation tools can not directly follow the hierarchical simulation approach used for VLSI, because the VLSI mixed-level paradigm is too narrow to address the needs of MEMS designers. In VLSI, the mixed-level approach is based on a single low-level description – circuits – and a single hierarchy of macro-models. There is no single low-level description in MEMS: designs involve a mixture of forces due to electrostatic fields, fluids, mechanical elasticity, etc. This lack of a single low-level description has two important ramifications: there is no organized approach either to generating or to coupling together different levels of MEMS representation.

MEMS requires a mixed-level approach, where both different physical systems and different levels of models can be coupled together in an organized fashion. In order to take an important step towards solving the mixed-level simulation problem, it is necessary to develop a software simulation system which will not only allow coupling between fluids, electrostatics, and mechanics, but will also allow a mixture of different physical regimes for the different energy domains. Once such mixed-regime approaches are developed, coupling to existing commercial circuit simulation tools like SPECTRE(Cadence) or SABER(Analogy), can provide full mixed-level simulation capabilities. In the rest of the paper we summarize our present efforts in developing algorithms for coupled-domain and mixed-regime simulation.

One approach to coupled-domain simulation is to use very general finite-element analysis approach. In such approaches, the unknowns in the various physical domains are represented by a sum of basis functions whose coefficients are determined by a Galerkin condition applied to the appropriate physical equations. The main short-coming of the finite-element approach is that it does not allow for individual selection of the most efficient simulation algorithms in each of the physical domains. For example, consider coupled electromechanics. The exterior field problem is most efficiently solved with accelerated boundary-element methods, but mechanical elastostatics is most efficiently treated using standard finite-element methods. Using finite-elements for both computations would be extremely expensive.

Another common approach to coupled-domain simulation is the relaxation scheme where the domains are solved separately and the solution is advanced iteratively until a self-consistent solution is found. The advantage of this iterative technique is that it allows the most efficient simulation algorithms to be used in each of the physical domains, and in addition, simulators can be coupled without being rewritten. The problem with the relaxation algorithm is that for a variety of applications, such as high-field electromechanics, the relaxation fails to converge.

Another approach that has been studied for coupled problems is the surface- Newton method. The key idea in this approach is to reduce the dimensionality of the coupled problem from 3-D to 2-D, where only the surface variables are involved in the coupled equations, and to apply a Newton method. For example, in coupled electromechanics, once the displacement of the structure surface is known, both the surface electrostatic force and the structure's interior displacements can be determined by decoupled electrostatic and mechanical analysis. Surface-Newton approach not only preserves the easy extensibility of the relaxation scheme but also eliminates the convergence problems encountered with the relaxation scheme. The tangent in the surface-Newton method is evaluated through the use of matrix-free conjugate-direction algorithm. Unfortunately, most simulation packages are not designed to allow for efficient computation of matrix-vector products as required in surface-Newton methods.

Our main point then, is that perhaps individual simulation packages should be redesigned, so that matrix-vector product computation is efficient, as this will allow for a ”plug-and-play” approach to coupled-domain simulation. Then, much more rapid progress could be made on coupled fluid-mechanics-electrostatics-magnetics-circuits problems.