An Environment for MEMS Design and Verification

S. F. Bart*, N. R. Swart, M. Mariappan, M. H. Zaman*, and J. R. Gilbert*

*Microcosm Technologies, Inc., 215 First St., Cambridge MA 02142 sbart@memcad.com
Analog Devices Inc, 21 Osborn St., Cambridge MA 02139 Nicholas.Swart@analog.com

ABSTRACT

A comprehensive CAD software environment is presented for MEMS/MST design. It includes MEMS mechanical schematic entry, parameterized cell layout generation, automatic lumped parameter macro-model generation, and LVS style error checking. These tools allow for flexible design including parameterized-cell layout generation. Lumped parameter macro-models can be generated automatically from 3-D, static, FEM/BEM based device simulations. The software environment also provides error checking between different design views. The design environment is illustrated using a differential capacitive-based accelerometer as an example. Efficient, dynamic simulations of this large example system with models based on true 3-D device behavior are shown.

Keywords: Microsystems, CAD, MEMS, MST, FEM.

INTRODUCTION

Much of the promise of microsystems technology (MST) or of microelectromechanical systems (MEMS), comes from the ability to intricately link a sensing device with signal conditioning circuitry in increasingly complex systems. (see [1], for example). Such complexity will require a comprehensive design environment, which includes design capture, system level simulation capabilities, and verification [2].

Current state-of-the-art VLSI design environments benefit from several decades of research, which has resulted in sophisticated tools and methodologies for rapid design of complex analog and digital integrated circuits. To a large degree, a top-down, modular approach is employed. A common set of elements found within these VLSI design environments are: (1) design capture, usually by means of a circuit schematic entry tool in the case of analog circuits, and a behavioral description language in the case of digital circuits, (2) simulation, using circuit simulators, or digital simulators, with appropriate device macro-models of transistor or gate behavior, (3) layout, whereby a design is transferred to a set of two-dimensional masks, either manually or automatically, (4) verification, whereby layout is verified against design intent, i.e. the captured design.

While VLSI design environments have reached a significant level of refinement, corresponding design environments for microelectromechanical systems (MEMS) are still in their infancy, with much work remaining to be done. Some progress has been made in certain areas, *viz.*, MEMS-specific enhancements of simulation approaches such as the finite element method (FEM), or the boundary element method (BEM) [3]. However, other essential components in the MEMS design flow still require additional research. Further, the co-integration of the VLSI and MEMS design environments has not yet been addressed.

In an effort to address these shortcomings, we present an integrated environment for efficient design of advanced MEMS devices. The design flow for the environment is shown in Fig. 2, the important components being a mechanical schematic entry tool, a structured layout editor, an automatic macro-model generator for system level simulation, and a layout verification engine.

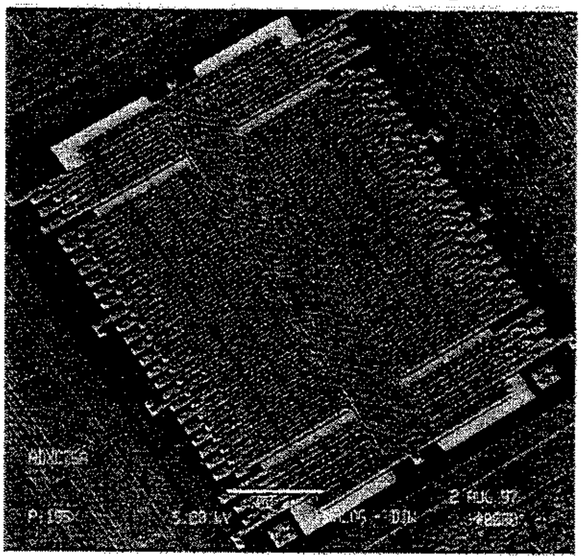


Figure 1. A SEM of the MEMS sensor portion of an Analog Devices' ADXL76 single axis 50-g accelerometer.

One of the key underlying features of the presented environment is the notion of reusable mechanical elements, which have associated system level models and structural generators. To this end, the design flow consists of two paths, one which is employed for electromechanical component generation, and the other which is employed for the design of a MEMS device. The various design flow components will be explained in the sections that follow.

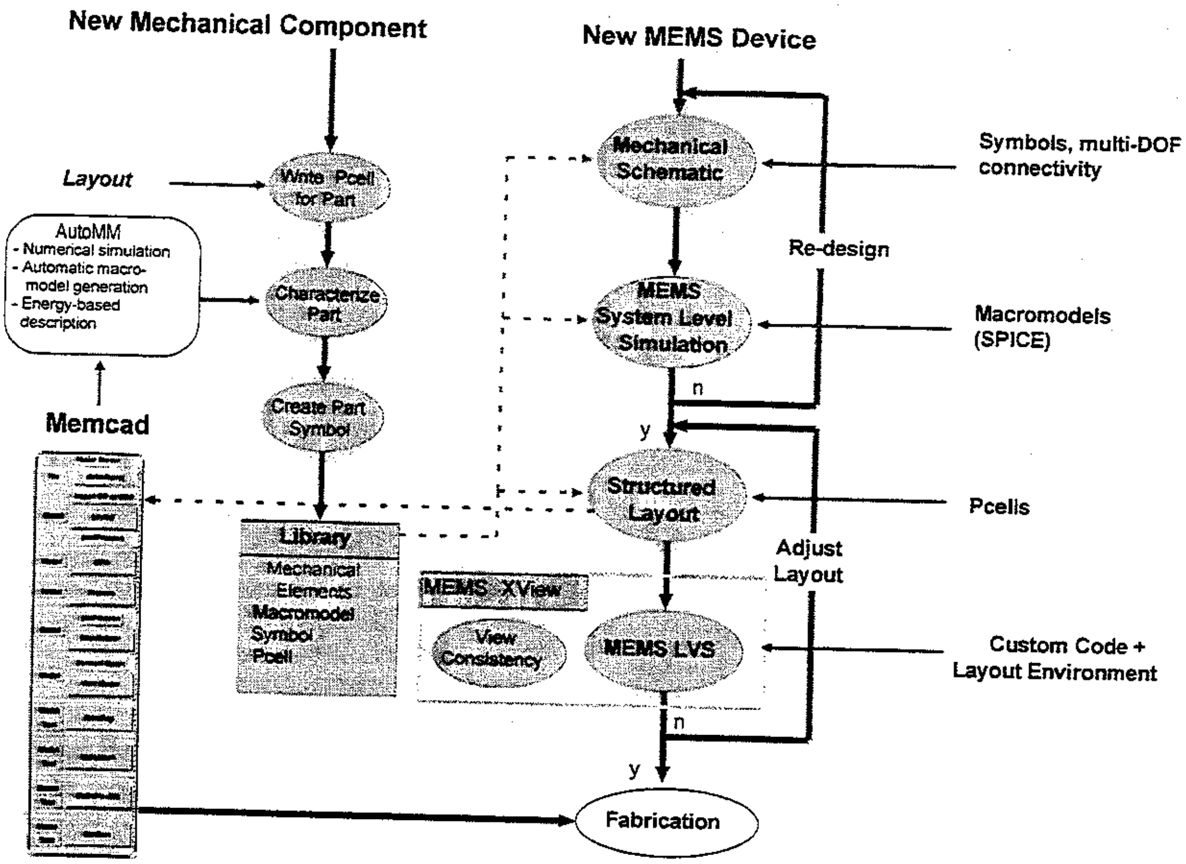


Figure 2. Design flow for the CAD design environment.

A production capacitance-based accelerometer (Analog Devices' ADXL76; see Fig. 1), is employed as an illustrative example. The device consists of a central moveable mass, which is supported by tethers on either end. The tethers support the mass, allowing it to displace under acceleration [4].

MECHANICAL SCHEMATIC

A mechanical schematic is used for capturing the design intent of a MEMS device. This lends itself very well to a hierarchical design methodology which VLSI circuit designers are familiar with. As shown in Fig. 2, the components instantiated in the mechanical schematic come from a library of re-usable MEMS components.

The electromechanical schematic of the ADXL76 is shown in Fig 3. The mechanical schematic for this system captured using a circuit schematic entry tool is shown in Fig 4. All the components have mechanical ports that are connected to each other via a mechanical bus (the thick wire in Fig. 4). Multi-degree-of-freedom mechanical values get passed between these ports. The conducting structures have electrical connection to the external circuitry via electrical ports and electrical wires.

By mapping position and force of the mechanical components to voltage and current, a SPICE netlist for the MEMS device can be generated from the schematic. The electromechanical macro-models for the primitive elements will be created using the Automatic Macro-Model Generator (AutoMM) described in Section 4. These primitive models can be combined with the circuit netlist to form a complete system level netlist.

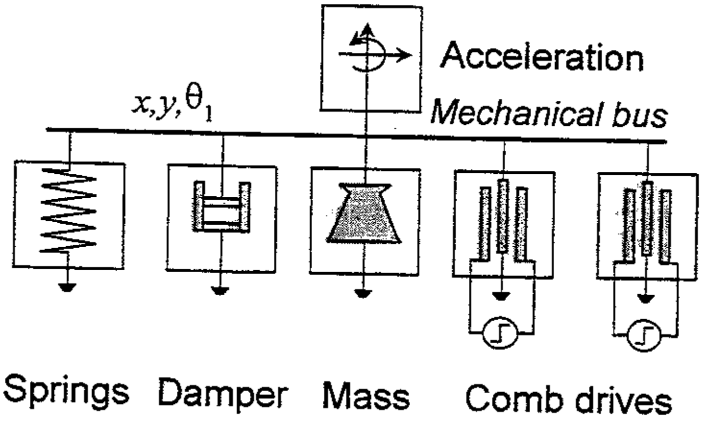


Figure 3. Diagramatic representation of the electromechanics of a capacitive accelerometer.

The system's design library consist of the primitives required for creating a MEMS device like mass,

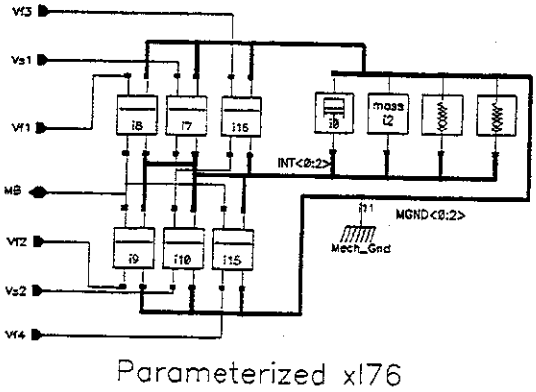


Figure 4. A mechanical schematic of the ADXL76 as implemented in the mechanical schematic.

springs, and interdigitated capacitive fingers. Each component in this library is parameterizable when it is instantiated on the schematic. Clearly, any given set of primitive types, even though they are parameterizable, will not be able to encompass the full extent of MEMS designs Therefor, it is important that the system allow new MEMS component primitives to be created quickly. The mechanism for creating such primitives is discussed in Section 4.

STRUCTURED LAYOUT

Parameterized cell (Pcell) generators provide designers a method of creating programmable layout cells for MEMS devices. A library of MEMS elements is created using Pcell generators. This provides designers a framework for doing structured MEMS design.

A Pcell generator takes parameters from the user and automatically generates a layout of the primitive component. Using the layout created by this generator, the part can be characterized and a macro-model created, as shown in Fig. 2. Simple analytical relations may be used to verify that the geometrical parameters used in creating the layout, like length and width, match the physical parameters in the schematic, like mass.

As an example of structured layout, Analog Devices' ADXL76 accelerometer was decomposed into its primitive lumped components: mass, springs, and capacitive finger cells. Pcell generators were developed for each of the components within the layout editor environment. When a new component is created, the geometrical parameters for the component are entered through a graphical user interface. The layout created using these generators is used as the input to the Automatic Macro Model generator. Fig. 5 shows the structured layout of the ADXL76. The mechanical connectivity between the different components in layout is established using ports. A connection is assumed when the ports of two elements overlap each other.

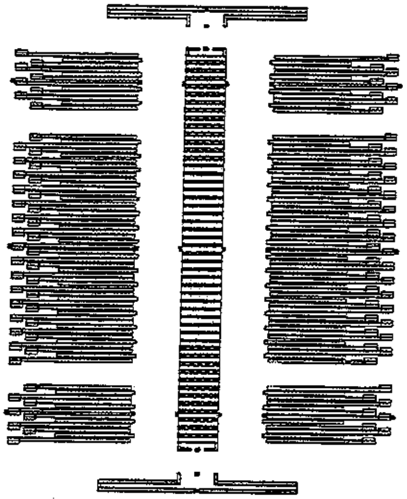


Figure 5. Pcell-based layout of the ADXL76 accelerometer. Note that various components have been separated from the central mass for illustration. In the actual layout, the ports (black squares) would be coincident, indicating a mechanical or electrical connection.

MACRO-MODEL GENERATION

The automatic macro-model generation module, AutoMM, is built upon the base computation engines in the MEMCAD software suite [5]. It uses the MEMCAD device creation tools and applies wrappers around the FEM/BEM solver modules to create multi-degree-of-freedom, high-order, lumped macro-models. The starting point for device-level 3-D simulations is a meshed 3-D solid model. These models can be built either directly by the designer using MEMCAD's automatic solid model building tools, or automatically from the masks generated by the Pcells described above. At present, meshing must be done by the user in either case. However, expanding the Pcell functionality to directly generate both a 2-D mask set and a 3-D mesh is relatively straightforward.

At present, AutoMM computes three types of lumped element. The mass, or moment of inertia element is computed from the geometry of the solid model. The spring element is generated from force versus displacement results from 3-D static mechanical (FEM) simulations. The electrostatic transduction element is generated from capacitance versus displacement results from 3-D static electrostatic (BEM) simulations. The reader is referred to Swart, et al. for more details [6]. The user has the ability to perform these computations on the entire MEMS device. Alternatively, if the device has physical features whose behaviors correspond to the model's lumped elements, the solid model can be broken into these physical components. Each component solid model can then be simulated separately in the appropriate physical domain. Examples are the folded flexures of the ADXL76, which provide all of the important compliance of the device and are easily isolated as separate physical components in the solid model. In addition, they do not exhibit important electrostatic behavior, so they need only be simulated in the mechanical domain. This partitioning can provide large computational savings. These computations can also be computed in parallel to reduce the over-all time of macro-model generation. Finally, a damping element is provided which requires the input of a damping constant by the user.

The AutoMM software module has been divided into several sub-modules that parallel the functions required to generate the macro-model. All of these sub-modules act on device data files called MBIF (Mem Base Interchange Format) files. An MBIF file is created from a meshed solid model. It contains both the mesh and material property data in a convenient way that facilitates transformation and simulation by computation engines that span multiple physical domains.

The first AutoMM sub-module performs a "global base transformation" on all device components that may exist in one or more MBIF files. These transformations allow the user to vary physical dimensions or material parameters such as gap separations between structures, angular orientations, lengths, thickness, density, elastic modulus ratio, material stress, etc. Note that these base transformations allow the user to generate models that explore the manufacturing variation space. The transformed models are then passed to sub-modules that perform electrostatic, mechanical, and inertial simulations using multi-DOF boundary conditions. The simulation data are then automatically fit to multi-degree polynomial equations (user specified, up to fourth order), which are functions of the degrees-of-freedom over which the device has been simulated. Fig. 6 shows the simulation data and the polynomial fit for a beam tether (spring) from the ADXL76. The polynomial fit coefficients are finally used in an energy-balance equation to create the device macro-model [6]. Although most of these steps are automated, user interactions and interventions have been allowed in a few cases to allow monitoring of the simulation process.

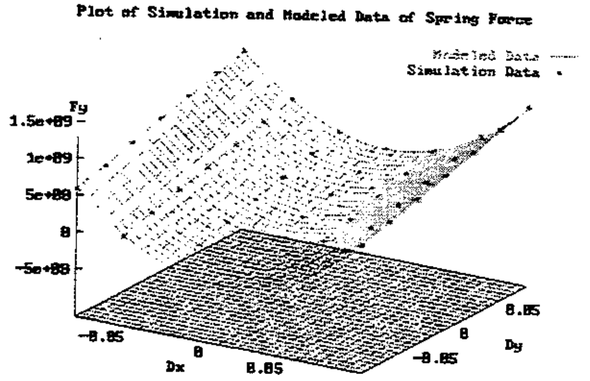


Figure 6. Force versus Displacement for a spring (tether) from the ADXL76 accelerometer. The plot shows two coupled DOFs: dy = sensitive axis of the accelerometer and dx = the transverse in-plane axis (microns). The dots are the simulation data and the lines are the curve fit. The force is in fN.

SYSTEM SIMULATION

The automatic macro-model generation module represents an essential component in a mature simulation environment for MST. Most analysis and simulation takes place at two levels: the "physics" level, in which one solves for the detailed behavior of the device in 3-D, and/or the "system" level, in which one solves for the behavior of the entire system. In the latter, the system is represented as a collection of components that are each described by compact or reduced order models (macro-models). Tools for working at the "physics" level are typically based on solving partial differential equations (PDEs) or integral equations, using finite element (FE), finite volume (FV), boundary element (BE) or related techniques. At the "system" level, one uses ODE solvers such as SPICE, or similar tools. Fig. 7 shows a diagram of these levels.

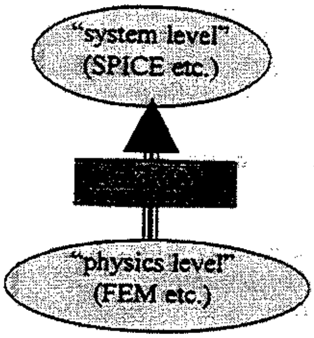


Figure 7. Macro-models connects the "physics" of the MST device to the "system" it fits within.

AutoMM extracts a "system level" model of an MST component by doing a controlled set of "physics" level analyses as described in Section 4. These computations are static simulations where the physical domains are uncoupled due to the assumption of lumped behavior. The degrees-of-freedom over which the models are extracted are the translational and rotational DOFs. Other DOFs, such as the modal eigenvectors, can act as the basis functions for such an extraction. The resulting macro-model is a multi-degree-of-freedom, high-order, lumped macro-model, which can be used with a system level ODE solver to perform transient and other dynamic simulations of the device behavior. This technique is much more computationally efficient than the case where the PDE solver is used directly in the dynamic simulations.

In addition to exploring device level dynamics, the SPICE-compatible model can be employed in a full chip-level simulation of the system with its associated control and readout electronics. The readout electronics can be simulated either at the transistor level, or at a higher level using lumped models for the various electronic sub-blocks. In either case, the pin-level behavior of a MEMS

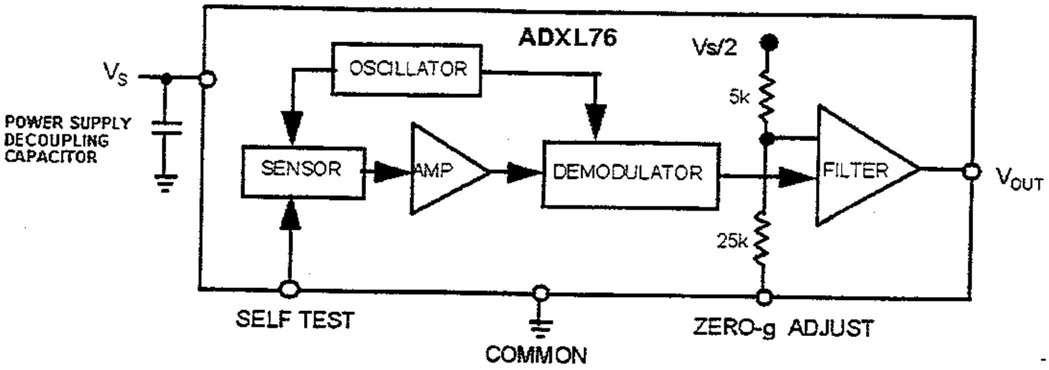
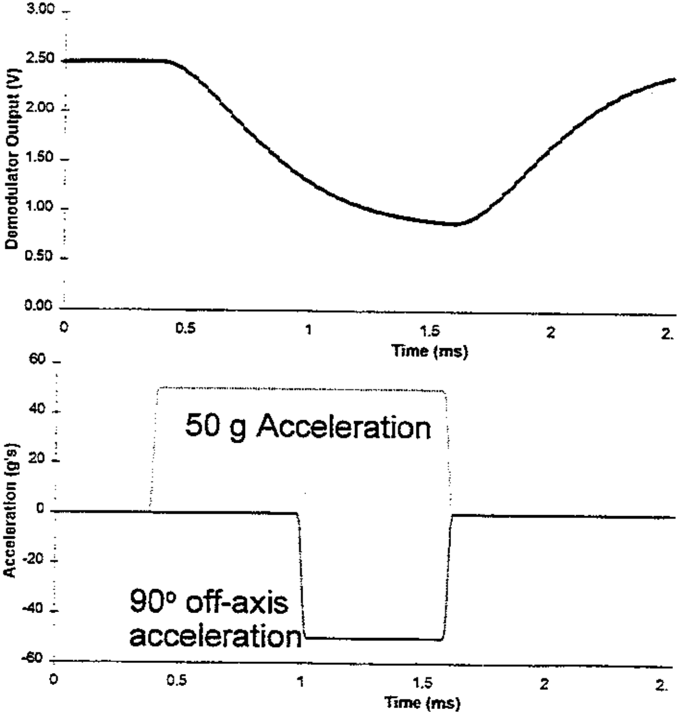


Figure 8. System level block diagram of the ADXL76 MEMS accelerometer.

component can be studied, verified and optimized, which is crucial for complex designs.

As an example, we present the results of a chip-level simulation of the ADXL76. In Fig. 8, a system diagram of the chip electronics is shown. The accelerometer is capacitive based; an external force in the plane of the device results in a differential change in the capacitive output. The capacitance is read by two out of phase signals, which are applied on either side of the differential sensing elements. The output signal of the MEMS device is fed through a switched-capacitor demodulator. The circuitry also includes various timing blocks for the switched capacitor network, and a collection of switches for performing self-test functions.

Figure 9. Simulated response of the ADXL76 under applied on-axis and out-of-plane acceleration.



The result of a coupled simulation of the ADXL76 electronics (transistor level) and the 2-DOF accelerometer model is shown in Fig. 9. Here, the output of the demodulator is plotted under an applied acceleration signal, which includes both on-axis and off-axis 50-g pulses. The low-level interaction of the highly non-linear electronics, and the electromechanical dynamics of the accelerometer, have been captured in a coupled simulation. The calculated sensitivity of ~35 mV/g is in excellent agreement with the measured nominal sensitivity of 36 mV/g for the production device.

If the ADXL76 were operated using a single ended capacitive readout, it would have unacceptable off-axis sensitivity. However, by employing a differential readout circuit, the off-axis response is negligible as can be seen in Fig. 9. There is no discernable change in the demodulator output when the off-axis 50-g pulse is applied. This agrees well with the measured behavior of the device, where, disregarding manufacturing variations, a similar behavior is observed. Inclusion of manufacturing variations in the automatically generated macro-models is being researched.

VERIFICATION

The ability to verify the integrity of the different representations of a MEMS design will play an important role in reducing errors in the design and the transition to manufacturing, particularly for complex devices. As shown in Fig 1, each element of the device has several views (i.e. Pcell, schematic symbol, macro-model and 3-D solid model). All the views of an element will have to be consistent with respect to port names and port sense.

To this end, a cross-view checking tool was developed which checks for consistency and dependency across the different MEMS device views. The consistency check confirms that all the views of a MEMS element have the identical port names and port sense. The dependency check confirms that the layout of a component has not changed since the macro-model for it was created. The date

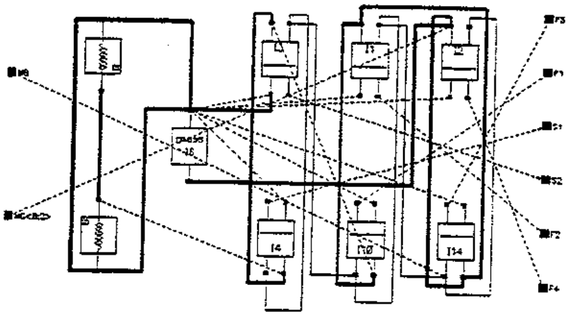


Figure 10. Connectivity diagram extracted from the ADXL76 layout.

stamps for the different views are verified to confirm they are synchronized.

A second essential aspect of design verification is validating design intent against layout. We have developed a tool that extracts electromechanical connectivity from a Pcell-based layout. The connectivity diagram of the ADXL76 layout in Fig. 5 is shown in Fig. 10. This view contains all the mechanical and electrical connectivity contained in the original mechanical schematic (Fig. 4). An LVS verification tool will be developed based on this connectivity generator. This tool will compare the connectivity diagram to the mechanical schematic to verify their equivalence.

Once the MEMS verification is accomplished, this can be combined with traditional circuit verification to achieve full chip-level verification.

CONCLUSIONS

We have presented a comprehensive environment for MEMS/MST design. It includes MEMS mechanical schematic entry, parameterized cell layout generation, automatic lumped parameter macro-model generation, and LVS style error checking. The design environment is illustrated using a differential capacitive-based accelerometer as an example. These tools allow for flexible design with automatic layout generation, efficient dynamic simulations of large systems with models based on true 3-D device behavior, and error checking between different device views.

In the future, the design environment will be extended to include rotational degrees of freedom for electromechanical systems. To this end, the required enhancements to the mechanical schematic, the structured layout tools, and the verification engine will be made. Further, the necessary tools and approaches for studying device behavior under manufacturing variations will be included in the design flow. This will result in a powerful MEMS design environment, which compliments those currently available for VLSI design, thereby enabling development of sophisticated MEMS devices with integrated electronics.

Acknowledgements

We gratefully acknowledge DARPA for partial funding of this work under the composite CAD program, contract number F30602-96-2-0290.

References

1. J. A. Geen, "A Path to Low Cost Gyroscopy," to be presented at the 1998 Solid-State Sensor and Actuator Workshop, Hilton Head, June 7-11, 1998.
2. S. F. Bart, "CAD for Integrated Surface-Micromachined Sensors: Present and Future," in the Proceedings of the Fifth ACM/SIGDA Physical Design Workshop, Reston, VA, April 15-17, pp. 72-75, 1996.
3. S. D. Senturia, "CAD for Microelectromechanical Systems," International Conference on Solid-State Sensors and Actuators, Transducers '95, Stockholm, June 26-29, 1995.
4. S. F. Bart, and H. R. Samuels, "Design Techniques for Minimizing Manufacturing Variations in Surface Micromachined Accelerometers," in *Microelectromechanical Systems,* DSC-Vol. 59, ASME WAM, 1996.
5. *MEMCAD 4.0 Manual*, Microcosm Technologies, Inc., 5511 Capital Center Dr., Raleigh, NC, 1998.
6. N. S. Swart, S. F. Bart, M. H. Zaman, M. Mariappan, J. R. Gilbert, and D. Murphy, "AutoMM: Automatic Generation of Dynamic Macro-models for MEMS Devices", Proceedings of the Eleventh Annual IEEE Micro Electro Mechanical Systems Workshop, Heidelberg, January 25-29, pp. 178-183, 1998