A Novel Method to Utilize Existing TCAD Tools to Build Accurate Geometry Required for MEMS Simulation

Nathan M. Wilson[1], Sam Liang[2], Peter M. Pinsky[1], and Robert W. Dutton[2]
Departments of Mechanical[1] and Electrical[2] Engineering
Division of Mechanics and Computation[1] and TCAD group[2]

CISX 332, Stanford University, Stanford, CA 94305-4075
email: nwilson@leland.stanford.edu

ABSTRACT

This paper details a technique that exploits domain decomposition and utilizes a combination of 1-D, 2-D, and 3-D process simulation to build physically accurate geometry of micro-electro-mechanical systems (MEMS) for simulation. The size and aspect ratios of typical MEMS structures differ significantly from those traditionally found in the VLSI community. This spatial stiffness makes it difficult to construct a geometry model using standard process simulation tools. The domain decomposition technique is an automated process that uses process flow, mask layouts, and a set of heuristics to determine which regions on the wafer require 1-D, 2-D, and 3-D simulation. The appropriate order of simulation is then performed and the results are automatically combined to create a useful geometry which can be used for simulation of behavior.

Keywords: micro-electro-mechanical systems (MEMS), geometry, process simulation.

INTRODUCTION

In the MEMS field, it is becoming widely accepted that representing the 3-D geometry of devices is critical for simulation based design. Many commercial and academic systems for MEMS analysis rely on purely geometric operations to create the structures. Notable examples include MEMCAD, Intellicad, and Solidis [1]. While tools exist for creating geometry for the VLSI community, there are unique challenges posed by MEMS. Modeling challenges include the size (and aspect ratios) of typical structures, residual stresses which create initial curvature in ideally flat devices, and effects of the materials used.

The typical dimensions of MEM switches (such as fabricated using the MUMPS process) are on the order of several hundred microns in length, a width on the order of 1/10 the length, and thicknesses of only several microns. These dimensions cause significant problems for deposition and etching process simulators such as SPEEDIE [2]. If full 3-D simulations are performed, it would take excessive amounts of computing resources including time and memory which exceed current limitations of software and hardware. In addition, due to some of the features of typical devices, much of the device does not gain geometric accuracy from 3-D simulation compared to using 2-D typical cross-sections. This means an intelligent process must be undertaken to reduce the massive amount of data from a complete 3-D simulation to the geometric representation of practical use for device simulation. In this paper a novel method is demonstrated which utilizes existing tools in a computationally efficient method to achieve accurate and practical geometry.

OVERVIEW

The process of building a solid model representing a deposited layer of material has been divided into three basic steps to reduce the computational cost and permit the use of widely available process simulators:

1) Decompose the domain into 1-D, 2-D, and 3-D simulation regions.

2) Perform the appropriate simulations in each region.

3) Combine the resulting simulations together to create a final 3-D solid model.

DOMAIN DECOMPOSITION

Starting with a process flow and the layout masks, a geometry is created using conventional geometric operations. When a more precise representation of a layer (such as a layer of polysilicon to be used as the main structural component for actuation) is desired, the geometry is decomposed into 1-D, 2-D, and 3-D regions of simulation (Fig. 3). Since it would be difficult and computationally intensive to use the 3-D geometry of the device to determine the regions of simulation, the etch masks are used along with knowledge of the process flow. There are four major steps in the domain decomposition algorithm:

1) Create a uniform two-dimensional grid to cover the wafer.
2) Classify each box as a 1-D, 2-D, or 3-D simulation region.
3) "Grow" the classified regions using heuristics.
4) Merging similar adjacent regions.

Step 1 - creating a uniform grid

First, a tensor product (rectangular) grid is created using the deposition thickness (defined as the total thickness of deposition on a flat surface of the current step) rounded up to the nearest integer as the grid size. To be precise, "grid size" actually refers to the height and width of each individual box, and thus the total number of boxes is given by:

$\eta = ceil(l\ w\ /\ ceil(g)^2)$ where: η = total number of boxes (integer), l = length (float), w = width (float), g = grid size (float), and ceil(x) = function returning the smallest integer not less than x.

Step 2 - classification of each box

The next step is to loop over the η boxes and determine the nature of the simulation required in each box (see Figs. 1 and 2). It is important to note that in the case when multiple etch masks have been used to create the surface topography, a simple superposition of the masks is used. The assumption here is that regions requiring higher dimensional simulation (2-D or 3-D) can only occur within a given distance (which will be called δ) of the edges of the etch masks. Physically, this makes sense because the effects of a feature, e.g. a stepup, will only be felt in the local area of the feature on the wafer and deposition a "reasonable" distance away will be unaffected. Based on numerical experiments and empirical results, δ is assumed to equal to the deposition thickness for the current step. In the case of a wafer that undergoes multiple (repeated) depositions utilizing the domain decomposition technique, δ is the sum total thickness of all of the relevant deposits. It is believed that this is a conservative estimate (i.e. upper bound) of the effects of a given feature and appears to be the case in practice.

Step 3 - growth of 2-D and 3-D regions

The next step in the process is to grow the 2-D and 3-D simulation regions appropriately. The best way to justify this step is by example. Take a hypothetical box from above which contains a corner and has been classified as a 3-D simulation region. Now, assume that the corner occurs at an extrema of the box, such as the lower left hand corner. By the classification procedure in step 2, the box may have as its left neighbor a 1-D box (see Fig. 4). This is incorrect because the assumption was that the feature (i.e. corner) has an effect in a region of influence given by δ. One possible course of action to handle this problem might be to query each 2-D and 3-D box to gain additional information about the exact feature contained within. The drawback is that this would require handling many special cases. A simpler procedure, which was implemented is to create a safety zone around each 2-D and 3-D simulation region.

In general, the etch mask edges may be in arbitrary directions. The current implementation handles only Manhattan style etch masks. Thus, for a 2-D box containing a "vertical" etch mask edge, its adjacent neighbor box to the left and right are converted to 2-D if either were 1-D. For a "horizontal" etch mask edge, its adjacent neighbor box above and below are converted to 2-D if either were 1-D. For 3-D boxes, all eight of the adjacent neighbors are converted to 3-D (see Fig. 4).

Step 4 - Combining adjacent 2-D and 3-D regions

Finally, adjacent boxes of the same order are combined to yield rectangular simulation regions of a given order. For the 1-D and 2-D regions this is a straight forward procedure. In the 3-D case, however, it is more complicated because initially separate 3-D regions might be touching after the growth detailed in step 2. An efficient contour tracing algorithm is used to find the largest possible 3-D region (i.e. bounding box) by combining interacting or nearly interacting 3-D regions (see Fig. 4). The major motivation to have the rectangular 3-D simulation regions is that it greatly simplifies several procedures (such as the pseudo-3D technique described below) without significant cost penalty.

PERFORMING THE SIMULATIONS

For the 1-D regions, the deposition thickness has been user specified in the process flow, and this results in planar surfaces in the 1-D regions. In the 2-D regions, a 2-D process simulator [2] is used to predict the geometry. Surfaces are created by effectively extruding the 2-D profile in the appropriate direction to yield a 3-D surface. For the 3-D regions, two methods are being explored. Currently, we utilize 2-D process simulations to create pseudo 3-D geometry as detailed in [3][4]. Since the size of the area that requires 3-D simulation has been minimized by the domain decomposition, the use of commercial 3-D topography simulators is also being investigated.

RECONSTRUCTION ALGORITHM

After the process simulations have been run to create the geometry, the results have to be recombined to create a continuous deposition surface. Currently, the method implemented is to create a solid from each simulation region and then union (Boolean addition) all of the solids together to create the final 3-D geometry (Figs. 5 and 6). The motivation for the current implementation is its simplicity. However, robustness issues (predominantly with tolerancing) within the solid modeler have increased the need and efforts to search for more alternatives in combining the data. One method which holds great promise being explored is to "sew" the surfaces together thus creating a boundary surface which can be used to create to final solid.

ACKNOWLEDGEMENTS

This work was supported under the DARPA Composite CAD program contract #F30602-96-2-0308-P001. The authors would like to thank Clare Thiem and Thomas Renz of AFRL for ongoing helpful discussions about the geometry needs of the MEMS community.

REFERENCES

[1] MEMCAD, http://www.memcad.com., Intellicad, http://www.intellisense.com, ISE, htpp://www.ise.com

[2] SPEEDIE, http://speedie.stanford.edu.

[3] Wilson, N. M., Hsiau, Z., Dutton, R. W., Pinsky, P. M., 1998, "A Heterogeneous Environment for Computational Prototyping and Simulation Based Design of MEMS Devices", proc. SISPAD, pp. 153-156.

[4] Wilson, N. M., Dutton, R. W., Pinsky, P. M., 1998, "Utilizing Existing TCAD Simulation Tools to Create Solid Models for the SBD of MEMS Devices", International Mechanical Engineering Congress and Exposition, Anaheim, Nov. 15-20.

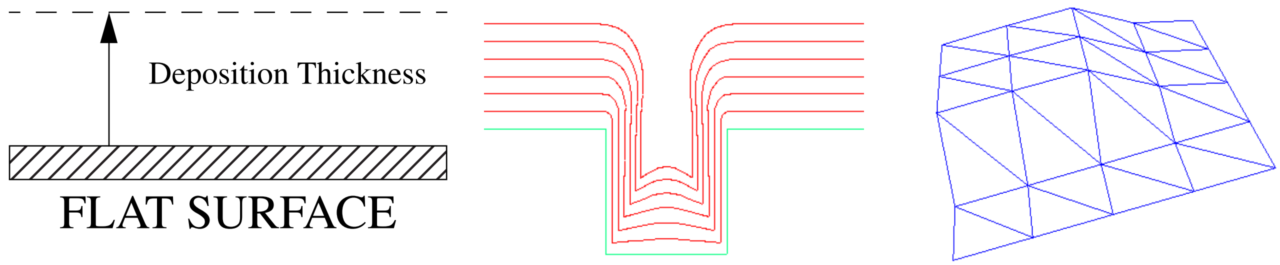

Fig. 1: The definition of process simulation regions. (a) 1-D. (b) 2-D. (c) 3-D. The 1-D regions require only a thickness specification, 2-D regions require SPEEDIE simulation, and 3-D regions need either the pseudo-3D technique or the use of a 3-D process simulation.

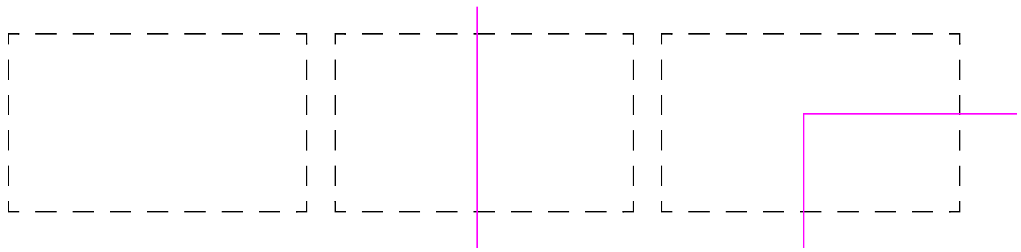

Fig. 2: The 3 possible cases in the domain decomposition of the masks. (a) no line segments inside of the box, indicates a 1-D region. (b) one line segment inside of the box, indicates a 2-D region. (c) two or more line segments indicating a 3-D region.

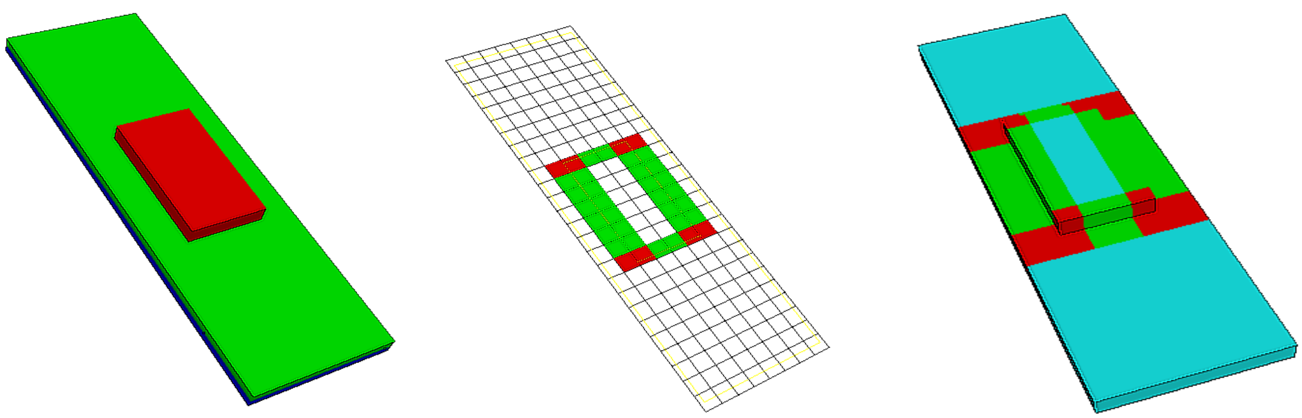

Fig. 3: The domain decomposition of a simple surface. (a) shows the original geometry. (b) domains based on etch masks used to create geometry. (c) the regions of simulation where cyan is 1-D, green is 2-D, and red is the 3-D domains.

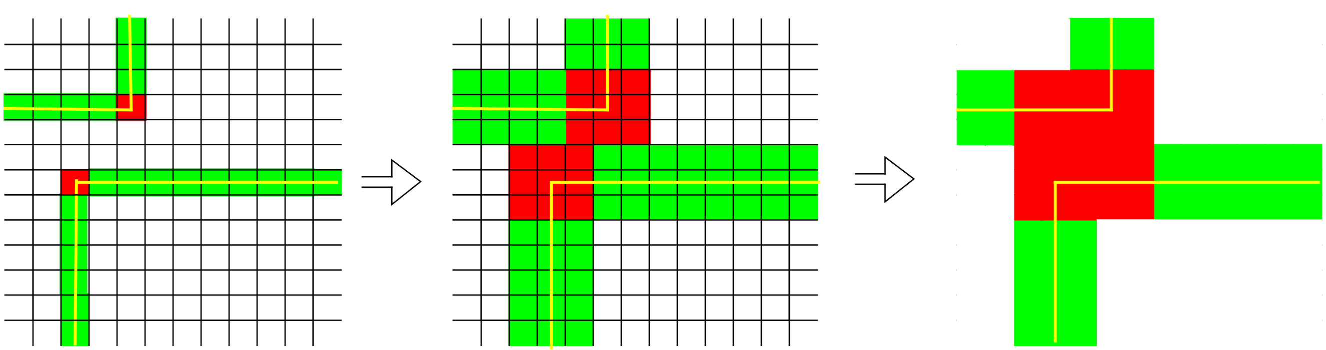

Fig 4. Domain decomposition algorithm. The yellow lines are etch mask edges, 3-D boxes are red, 2-D boxes are green, and the rest are 1-D boxes. (a) On the initial grid, classify the boxes. (b) Grow the 2-D and 3-D boxes. (c) Merge the 2-D boxes and 3-D boxes into 2-D regions and 3-D regions, respectively. The remaining white area is the 1-D region merged region.

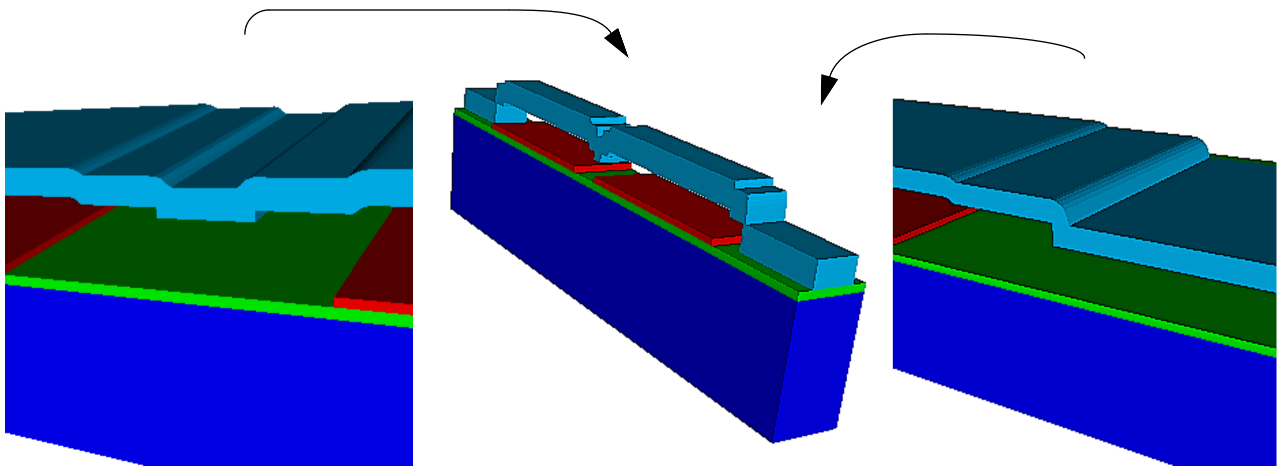

Fig. 5: Two electrode switch built using the domain decomposition technique utilizing a combination of geometric etches and physical depositions. (a) A close-up of the center of the switch. (b) 3-D view of the geometry. (c) The stepup.

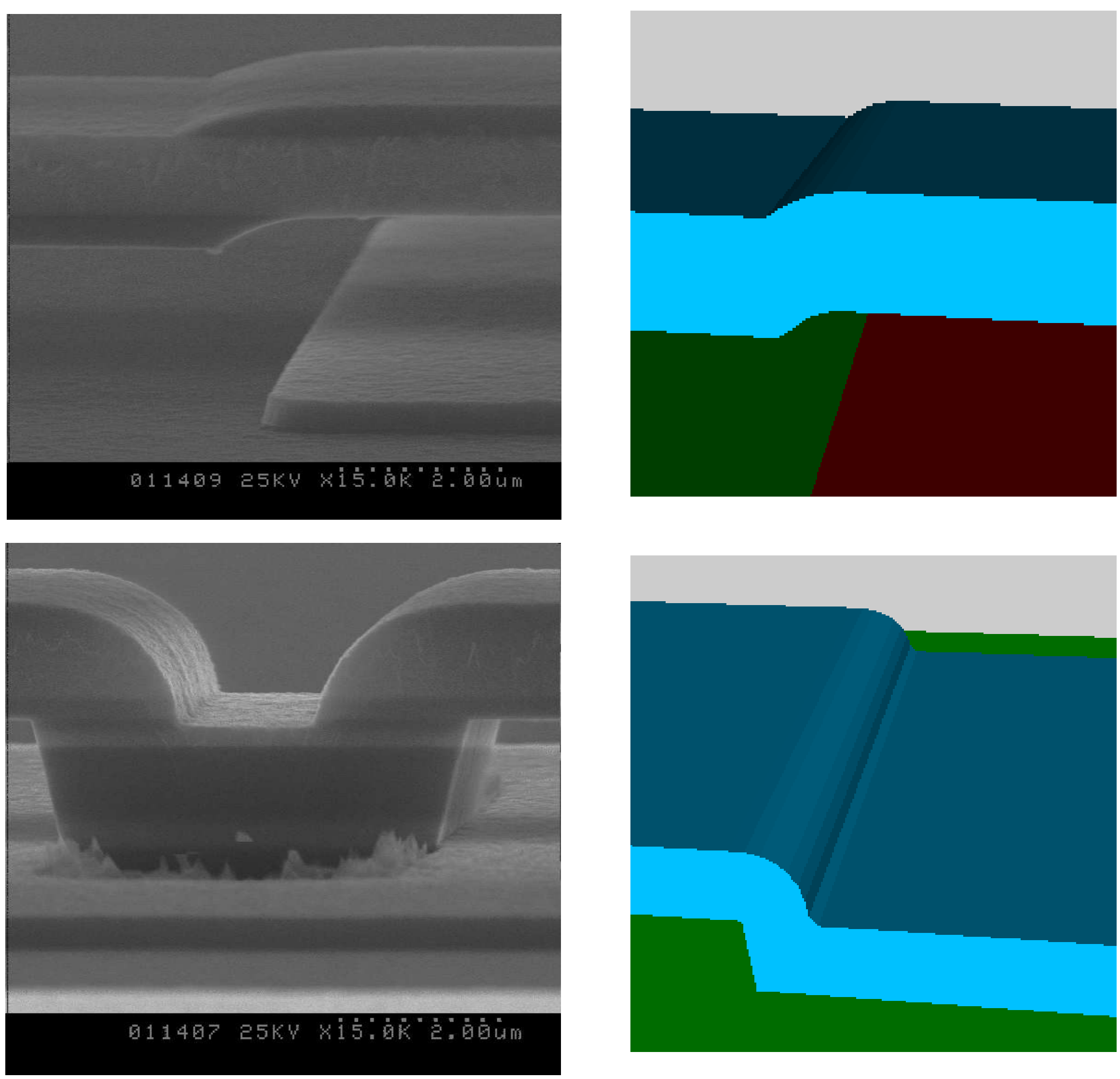

Fig. 6: On the left are two SEMS of an actual switch fabricated in the MUMPS process (see E. K. Chan et. al., MSM99), while on the right are two images of the solid model (see also Fig. 5). Notice the effects of conformal deposition on the final geometry.